

MTX User Club Deutschland

Info 9
30. 09. 1985

Zweck: Austausch von Tips & Tricks u.s.w. und Hilfestellung bei allen möglichen Problemen.

Programme (nur Selbstgeschriebenes): Tausch von kurzen und einfachen Routinen. Besprechung von guten Programmen damit der Autor diese dann an Clubmitglieder verkaufen kann. Programme an uns schicken, und wir liefern Verbesserungshinweise, Besprechung, ...

Mitglied kann jeder werden! Keine Aufnahme oder Beitragsgebühr!

Verpflichtungen: Einsendung unseres Fragebogens (liegt Info's bei)

Bitte: Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, Programmen, Hinweisen auf preiswerte Hard- und Software, und was noch so zusammenkommt und andere interessieren könnte.

Club-Info, unser Blatt, verschicken wir ca. 6-wöchentlich. Inhalt ist alles was uns über den MTX/FDX (ohne Copyright) in die Hände fällt.

Kosten: Wir berechnen ausschließlich Selbstkosten (Porto, Verpackung, Vervielfältigung, Datenträger, ...).
Wir verschicken **nichts**, wenn's Guthaben nicht reicht! (s.u.)

Da wir unseren Steckbrief nicht nur gegen Freiumschlag verschicken, ziehen wir denen, die ihn geschickt bekommen dafür DM -.70 vom Konto ab. (Einspruch ist selbverständlich jederzeit möglich.)

Geld/Konto: Für jedes Mitglied führt Herbert Herberg ein Konto, von dem die entstehenden Kosten jeweils abgehen. Der Kontostand wird bei **jeder** Sendung mitgeteilt (er steht über der Anschrift), und kann selbverständlich jederzeit erfragt werden!

Einzahlungen bitte möglichst auf's Club-Konto: (oder V-Scheck)
(Absender! incl Name und Anschrift nicht vergessen!)

Postgiroamt Hamburg, BLZ 200 100 20,
Herbert Herberg Sonderkonto C, Nr. 3480 00-200

Kontaktadressen:

Herbert Herberg
Sonnenau 2
2000 Hamburg 76
(040) 200 87 04

Thomas Pflaum
Leipziger Platz 1
8500 Nürnberg 20
(0911) 51 35 21

Frank Bueschler
Am Ochsenzoll 3
2000 Norderstedt
(040) 52 77 581

Moin, moin!

Sicherlich seid Ihr alle daran interessiert, was sich auf dem Clubtreffen des anderen MTX User-Club Deutschlands getan hat. Das wohl wichtigste und erfreulichste Ergebnis ist die geplante Fusion der drei uns bekannten deutschen MTX-Clubs. Der Termin hierfür ist der 1. November 1985. Das nächste Info wird also das erste gemeinsame sein.

Hier nur kurz die wichtigsten Änderungen: Einige Mitglieder erhalten in Zukunft das Info zum halben Preis, wir haben mehr Ansprechpartner, keiner muß mehr in mehreren Clubs Mitglied sein, weil er nicht weiß wohin. Ansonsten habe ich einige Mitglieder endlich mal persönlich getroffen und einiges an Software wurde vorgestellt.

Wie schon mit Thomas Pflaum geplant, wollen wir demnächst Disketten und Cassetten mit Gratis-Software sowie Public-Domain (d.h. CP/M ohne Copyright) verschicken. Die Datenträger werden nach Themen gruppiert (z.B. BASIC-Spiele, Turbo-Pascal, ...). Dazu mehr auf der Seite zum Thema Fusion weiter hinten.

Das Diskettenkonvertierungsprogramm **COMPAT** gibt es von Memotech direkt aus England! Preis ca. 85 Pfund. Wendet Euch ggf. für einzelne Konversionen an Frank Bueschler.

Mitglied sein bedeutet nicht, daß Ihr das Info beziehen müßt. Das ist für mich auch kein Problem, da wir zwei Nicht-Infobezieher im Club haben. Wer also kein Info bezieht steht in der Liste und ihm entstehen keine Kosten! Genaugenommen haben wir wegen unzureichenden Kontoständen bei einigen Mitgliedern (nein, Du nicht - Du hast ja Info 9) mehr Nicht-Bezieher.

Auch mir ist es aufgefallen, daß unsere Mitgliederliste explodiert, was mich sehr freut und das Info aufbläht. Nun ersteres hoffe ich, es geht rapide weiter - zweiteres habe ich behoben. I.a. werde ich nur einen Update der Liste ins Info setzen mit den Namen der Neumitglieder und denen mit neuen Adressen und denjenigen, die Software anbieten.

Das das erste Inhaltsverzeichnis nicht gerade berühmt ist wissen wir auch, aber Thomas hat ziemlich lange überlegen müssen, was als Stichpunkte, etc. sinnvoll ist. Daraufhin haben Uwe Beythien und ich und hingesetzt und versucht aus Thomas' Tabelle ein hoffentlich besseres und lesbareres Inhaltsverzeichnis gebastelt. Nun stehe ich vor dem Problem: rein ins Info - oder nicht. Da wir im November ein Zusatzinfo mit einigen anderen Tabellen veröffentlichen wollen (s.u. bei Fusion) habe ich beschlossen, daß ich in diesem Info nur zwei Teile dieses Verzeichnisses (BASIC und Korrektur) abdrucken werde. Bitte schickt mir möglichst rasch Verbesserungsvorschläge!!! und natürlich Korrekturen. (Wie z.B. das doppelte r in 'Korekturren').

Telefonisch bin ich zu folgenden Zeiten erreichbar:

Dienstags 19.30 - 21.30 und Samstags 9.00 - 12.00



P.S.: Eine große Bitte

Ich freue mich immer wieder über schriftliche Fragen, und beantworte sie auch gerne - wenn auch wegen der Menge hin und wieder knapp. Ich möchte **Euch alle bitten** solchen Fragen einen adressierten frankierten Rück-Umschlag beizufügen, damit ich es einfacher habe (sonst muß ich immer Eure Adresse raussuchen, und die Portokosten abbuchen). Auch auf dem Brief sollte Euer Absender vermerkt sein!!

P.P.S.: Im Info verwendete Abkürzungen :

dSM = dunkle Stunde von Memotech !!!!!!!

V o n F r ü h e r / H a r d w a r e / A n g e b o t e

Bisher erschienen und noch erhältlich: (Herbert Herberg)

RDM-Unterlagen (32 Seiten, 8.-), Info 1-6 (ca. 250 Seiten, DM 50.00)
Info 7 (11.96), Info 8 (65 Seiten, DM 11.05).

Das neue Info gibt's automatisch! (Wenn's Geld auf dem Konto reicht!)

Hardware

Zeichensatz-PROM mit eckigen Klammern und echten Oberlängen für die
80-Zeichenkarte (s. Info 6): DM 30.- bei Herbert Herberg.

Tastaturentprellung (d.h. Auseinandernehmen und Zusammenlöten) macht
Klaus Korgler für DM 30.-

Einbau von Diskettenlaufwerken DM 50.- + P&V, Herbert Herberg

Aufrüsten von RS 232-Karte DM 50.- + P&V incl. Bauteile, Herbert
Herberg

Aufrüsten von 32k-Karte mit 16 mal 4164 DM 30.- plus P&V und Kosten
für die IC's, Herbert Herberg

PROM für 32kKarte -> 128k DM 20.-, Herbert Herberg, bedingt lieferbar

Aufrüsten MTX 500 auf MTX 512 DM 60.- + P&V + Speicher-IC's, Frank
Bueschler

Professionelles RS-232 Kabel 150 cm, 19 polig, DM 30.-, Frank
Bueschler (Verwendungsfähig für dataphon s 21 d)

K o r r e k t u r - N a c h t r a g

Info 8, Seite 14 unten habe ich (Herbert Herberg) mir nichts Dir
nichts einige ATTR-Kombinationen angegeben (ATTR 2, und ATTR 3,).
Dies ist Murks! Ich habe es aus dem Handbuch übernommen - und ge-
glaubt.

Der Cursor-Geschwindigkeitspatch von Hagen Wenzek im Info 9 verwendet
Hex-Zahlen.

Im Info 6 war Seite 8 = Seite 9. Die so fehlende Seite ist im Info 7
längst erschienen.

Das FDX-Handbuch gibt einige Portadressen feil. Diese sind alle HEX!!

Hisoft-Pascal-Karte

verkauft J. Marquart für DM 150.-

EPSON SD 521

verkauft Uwe Beythien für DM 300.-

MTX 500 + DMX 80

MTX 12, DMX 4 Monate alt, verkauft J. Adamczak für DM 1000.-. Wenig
benutzt, einwandfrei!

MTX-Kompendium

ist ein von C. Löhrmann geschriebenes (FDX-)BASIC Handbuch als Ergän-
zung zum Memotech-Handbuch, aber wesentlich vollständiger und mit In-
haltsverzeichnis/Index. Viele Memotech-Fehler korrigiert. Die dritte
überarbeitete Auflage wird jetzt verlegt. Preis: DM 25.-

Hardware-Angebote

Wer irgendwo interessante Angebote für MTX-verwendungsfähiges sieht,
bitte notieren und Frank Bueschler schreiben. Frank hat eine Datei mit
solchen Informationen angelegt, und will diese immer auf dem neuesten
Stand halten. Natürlich genügt auch ein Hinweis an mich! Wissen müssen
wir Firma, ggf. Anschrift, Produkt, Preis, wo war die Anzeige. Danke!

Programme

Preise sind netto, d.h. ohne Datenträger und P+V (DM 8.-).

Die mit einem * gekennzeichneten Programme gibt es auch als Listing (DM -.30 je Seite); die mit # sind im Info gelistet. Die Programme mit einem ! brauchen den 80-Schirm in der FDX und die 40-Zeichen-Grafik.

Eine Zahl am Ende der Zeile ist die Nummer des Info's mit Besprechung. Ein (C) bedeutet Copyright, d.h. Kopierverbot!

Bitte die Programme bei dem, unter dessen Namen sie stehen bestellen!

Andreas Viebke (keine Nachnahme - wird zu teuer)

- 16.- Flugsimulator für Linienmaschinen (C) -> 2
- 12.- # D/E Zeichenvergrößerung und 40 Zeichen auf VS 4 (C) -> 4
- 20.- Schach-Editor (Schachpartien komfortabel auswerten) (C) -> 7
- 5.- Kleiner FDXB-Disassembler mit Druckerausgabe (C)
- 42.- XB (Extended FDX-BASIC) (C) -> 7,8
- 15.- XSPool (Druckerspooles unter XBASIC) (C) -> 9
- 15.- XD/E (Display-Enlarge unter XBASIC) (C)-> 9

Frank Dersewski

- 17.- CP/M-Programme: MDM712 (Modem), FORTH, DU (Disk Utility), RTTY (Teletype), ... -> 8

Michael Möwe

- 5.- PRT.COM (Druckerinitialisierung mit Zeichensatz) (C) -> 7
- 1.- HARD.COM (VS4-Hardcopy unter CP/M) -> 8

Kurt-Bernd Rohloff

- 0.- SETPRT.COM (Druckereinstellung) -> 6

Michael Köster

- 0.- MDS-Diashow-Hauptprogramm incl. 8 Grafiken -> 8
- Grafiken zur MDS-Diashow: -> 8
- 0.- Brigitte, Statue'', Eisvogel, Bauernml, Goofy''', Memotech, Telephon, Zebra''''
 - 1.- Wallys'' (C)
 - 1.- SHShadow (C)
 - 2.- Mexi'''' (C)
 - 3.- Hypnosis (C)
 - 3.- Schloss' (C)
 - 3.- Corsa'''' (C)
 - 4.- Kugel'''' (C)

Thomas Pflaum

- ?.- VS 4 - Grafik (ein kleines Grafikpaket in Turbo-Pascal)

Detlef Lütticke

- ?.- Textverarbeitung auf Cassette für MTX 500

Manfred Flume

- 5.- Textverarbeitung auf Cassette für MTX 512

Gunter Ebert

- 20.- DISCII (Terminkalender, Karteikarten, Statistik, FDXB) -> 9

Programme / Wer tut Was

Olaf Krumnow

5.- TURBO-Pascal-LISTER (C) -> 7
 demnächst: TURBO-Window, TURBO-Garfik,
 guter Z80-Monitor (ersetzt DDT, VDEB, ...)

Herbert Herberg: (Ich tausche auch und gebe Mengenrabatt!)

- 11.- RAM-Disc V.1 mit Druckerpuffer,.. (C) Bernd Preusing -> 5,7
- 3.- Fastcopy komfortables COPY (C) Herbert Herberg -> 6
- 0.- Disketten-Konvertierung TA-PC -> MTX
- 0.- FORMATS.COM / CONFIGS.COM (IBM-Format auf 5'') s.u.
- .- PIANO (Klavierprogramm von W. Tesch, Listing s.u.) -> 9
- 2.- Hbigcopy: DMX80 Hardcopy mit Vergrößerung um den Faktor 2
- 4.- # Charactergenerator incl Zeichensatz
- 1.- * Character-Designer
- 2.- BASIC-Tools (RENUMBER, FKEY) -> 9
- 1.- Matrizen (Mathematisch, Matrix-Operationen, BASIC)
- 1.- Musik (Musik unter BASIC)
- 3.- Biorhythmus
- 20.- Quibic (3D-Grafik 4 gewinnt) (C) Herbert Herberg -> 6,7
- 4.- Labyrinth (durch ein Labyrinth hindurchfinden, 3D)
- 2.- Lunar (Mondlandung, gut)
- 5.- Vier-Gew (Vier gewinnt)
- 4.- Missile (Städte vor Zerstörung schützen, ASSEMBLER)
- 4.- War-Plan (Flottenvernichtung)
- 2.- Othello (Brettspiel)
- 2.- Boot (Schiffeversenken)
- 3.- Brio (Geschicklichkeitsspiel: Kugel durch Labyrinth)
- 3.- Liner (plötzlich auftauchenden Linien ausweichen)
- 3.- Miner (Gold Berg finden)
- 2.- * Breakout (Mit Ping-Pong-Verfahren Steine zerstören)
- 2.- * Mamind (Mastermind = Kombinationsraten)
- 2.- * Jigsaw (Puzzle)
- 2.- Sandburg
- 6.- ! Railroad (Schienen legen mit bis zu 9 Spielern)
- 6.- ! Pferde (Pferdelauf mit mehreren Spielern)
- 2.- # Willie (Ein gefräßiger Wurm)

Wer tut Was

Allround:	H. Herberg, C. Löhrmann
(FDX-)BASIC	A. Viebke, H. Oppmann, V. Griener
CP/M System	B. Preusing
CP/M	K. Rohloff
Modem, Funk	F. Dersewski
Steuerungsprobleme	H. Riebesehl
Turbo-Pascal	D. Krumnow, B. Preusing, T. Pflaum
Grafik	M. Köster, A. Schmitt
Einkauf und Industriekontakte	F. Bueschler
Simulation (ökologieRaumfahrt)	U. Popella
Platinen	D. Harms

Wer sich auf dieser Liste fehlt am Platz oder vermißt fühlt ...
 schreibe mir. (Bitte nur ernstgemeinte Zuschriften, d.h. Ihr solltet
 im genannten Bereich "firm" sein, und es soller Euer Hauptinteresse
 bzgl. MTX sein).

Software / Tips & Tricks / Club**Kommentare zu einigen Programmen**

Quibic ist Monitorfähig, d.h. geändert. Update: ja!
 DISCII von Jens Ebert ist ein recht umfangreiches und Menügesteuertes Terminkalender, Karteikarten, Statistik und Grafik - Programm, dessen Prototyp mich seinerzeit überzeugt hat.
 Die Textverarbeitungsprogramme von Manfred Flume und Detlef Lütticke sind für die nicht-FDX-ler gedacht, wobei Manfreds 64kB braucht. Beide Programme kann ich empfehlen.
 VS 4 -Grafik unter Turbo von T. Pflaum ist recht umfangreich mit einigen Beispiel-Bildern. Leider verwendet Thomas nicht den MDS-Standard von Michael Köster!!!
 Der Prototyp von Olafs TURBO-Window ist sehr beeindruckend!

Programme & Disk/Cassetten, die Ihr verschickt

Bitte verseht alle Programme mit einer Dokumentation. Es macht sicherlich auch Spaß bei einigen Programmen herauszufindeln, was, wie, wo, warum (nicht) ...?? Bei Spielen ist es das sinnvollste vorneweg ein paar Noddy-Seiten zu setzen (oder auch legen). Umfangreichere Progs. sollten ein .DOC-File dazubekommen!
 Das Diskettenformat sollte **immer** vom Typ 03 sein (320kB) und man/frau die Disketten in Alufolie einrollen.

Tips & Tricks

NewWord: Automatische Seitennummerierung unten auf der Seite kann man mit dem Kommando .op (in der 1. Spalte) ausschalten.
 NewWord Druckerkommandos ^P^Q,.. ^P^R können im Text definiert werden, d.h. ohne NWINSTAL: Für ^P^Q: .xQnnnnnn, wobei an Stelle der nnn der zugehörige Code als Folge von HEX-Zahlen, wobei keine Längenangabe hinzu muß, und keine Leerzeichen verwendet werden dürfen. Für ^P^W mit .xW. Um Elite einzuschalten will der DMX 80 die Sequenz ESC P O, d.h. als HEX-Zeichen 1B 50 00. Das Kommando wäre damit für ^P^R: .xR1B5000. Aber wie auch bisher funktionieren diese Kommandos nur beim DRAFT-Druckertyp richtig. (Willi Gertel)
 NewWord BDOS Err: Bad Sector beim Abspeichern - und dann auch noch ohne .BAK-Erstellung von NewWord... Auch bei Unterdrückten Backups legt NewWord diese an, löscht sie aber nach **erfolgreichem** Abspeichern. Also RESET, und die .BAK-Datei umbenennen.
 FDXB Dateizu-Direktzugriff, d.h. Lesen (DISC INPUT) und Schreiben (DISC PRINT) akzeptiert nur ein Characterfeld! Dazu: DISC OPEN mit Typ "R".
 BDOS Err: Bad Sector erlaubt zwei verschiedene Reaktionen: BRK = Abbrechen, RET = Akzeptiere die erhaltenen Daten. So kann man ggf. noch große Teile einer defekten Diskette mit PIP retten, indem auf BDOS Err mit RET geantwortet wird.
 Diskettenlaufwerke könnt Ihr mit Schraubenzieher (um das Gehäuse zu öffnen), Kuh-Tip und Isopropylalkohol reinigen. Netzstecker ziehen! Vorsichtig dann mit dem getränkten Kuh-Tip gewaltlos die Köpfe abreiben.
FDX will nicht booten! Man sieht herrliches Schneetreiben auf dem Monitor. Hans-Dieter Müller hat eine Fehlerquelle entdeckt (**dSM**)! Die 32k- und RS232-Karten haben am Rand Massekontakte, die mit dem Gehäuse verbunden sein sollen - und die Platinen können hervorragend schlackern. Also herausnehmen, und die beiden Randkontakte **dick** belötzinnen (bis die Karten nur schwer hereingehen).

CP / M - Reparatur**An alle CP/M 2.2 - Benutzer** (Bernd Preusing)

Ein weiteres Hoch auf Memotech! Diese Firma hat es wieder einmal fertiggebracht, uns ein Kuckucksei ins Nest zu legen. Nur daß der Kuckuck eher einer Wanze (bug) ähnelt, seit Jahren durchs alte, aber doch ganz brauchbare Betriebssystem flattert und eigentlich vom Verteiler (hier MEMOTECH!!!) korrigiert sein sollte.

Da wir ja Kummer gewöhnt sind, machen wir uns **gleich(!!)** an die Entwanzung, aber erst will ich den Fehler kurz beschreiben; wer es ausführlich wissen will, muß zur c't 9/85 Seite 96ff greifen (sollte sowieso jeder abonnieren, Spitzenblatt für Freaks!).

Dieser Fehler tritt zwar recht selten auf, dafür schlägt er allerdings um so härter zu. Nämlich genau dann, wenn ein File geschlossen wird, in das Daten via random access geschrieben wurden und das danach genau so lang ist wie vorher. In diesen Fall findet beim Schliessen kein Directory-Zugriff statt, und es kann sein, daß das Bios noch einen halben physikalischen Sektor (128 Bytes) in seinem internen Block-/Deblock-Puffer hat, der einfach nicht auf die Diskette geschrieben wird.

So ein Fall kommt doch recht häufig vor, z.B. bei Turbo-Pascal im Programm TINST, von normalen User-Programmen gar nicht zu reden (auch dBase!).

Nun meint wahrscheinlich Memotech, diesen Fehler brauchte sie nicht zu korrigieren (**dSM**), weil unser Bios ja so unheimlich trickreich diese Hürde umgeht. Pustekuchen!! Unser Bios schreibt nämlich bei einem warm boot noch vorhandene Rest-Sektoren auf die Diskette, weil es den einen angefangenen Sektor loswerden muß, bevor es die Systemspuren lesen kann. Und was ist, wenn zwischendurch die Diskette gewechselt wurde? Dann ist nicht nur eine, sondern sind gleich zwei Scheiben im Eimer, weil der alte Sektor auf die Neue kommt!

Diese Bios-Macke mindert zwar die Wahrscheinlichkeit, daß der CP/M-Fehler auftritt, tritt jedoch besonders gerne in Aktion bei Programmen, die das BDOS umgehen, z.B. Direkt-Kopierer, Disk-Utilities etc.

Das einzige Mittel dagegen ist, den folgenden Patch zu machen und ansonsten zu beten oder ein neues Bios zu schreiben. Über Letzteres mache ich mir schon länger Gedanken; abwarten.

Durch den Patch müssen alle MOVCPM.COMs und alle Systemspuren (d.h. auf jeder Diskette!!) korrigiert werden:

```
DDT MOVCPM.COM          alt: 1CD2:   0D 0D C2 DF 12
(ändern mit 'S1CD2')   neu: 1CD2:   00 00 21 00 00
dann SAVE 40 MOVCPM.COM und alle vorhandenen Kopien überschreiben
(z.B. mit PIP).
```

Die Systemspuren kann man nun durch **SUB CPMGEN 58 B:** (recht aufwendig, aber ich habe Euch schon 'mal geraten Euer CP/M auf 58k umzustellen, dann hat man eine vernünftige TPA und hinter dem System noch genug Platz für z.B. RAM.COM) ändern.

Jetzt habt Ihr ein korrigiertes System, das Ihr mittels SYS oder SYSCOPY auf alle Systemdisketten bringt (dann sind aber alle STARTUPs gleich!).

Club - Fusion

Bislang gab (und genaugenommen gibt) es drei MTX-Clubs in Deutschland:

1. Herberg, Pflaum, Bueschler
2. Löhrmann, Griener, Hoffmann
3. Harms

Am **1. November** gibt es nur einen MTX User-Club Deutschland, natürlich dann mit uns allen. Um die Fusion braucht Ihr Euch nicht zu kümmern - wer in mehreren Clubs ist wird natürlich nicht mehrfaches Mitglied.

Im November werden wir ein Zusatzinfo herausbringen mit folgendem

- Inhalt:
1. Gesamt-Mitgliederliste
 2. Gesamt-Softwareliste, kommentiert
 3. Statuten
 4. Gesamt-Inhaltsverzeichnis

Dieses Info enthält keinen Fragebogen, wie ihn auch keine weiteren enthalten werden, da der Fragebogen durch einen Anmeldebogen ersetzt wird. Dem nächsten Info werden wir diesen Anmeldebogen beilegen mit der Bitte ihn ausgefüllt an uns zurückzuschicken. Darum werden auch "Schon-Mitglieder" gebeten.

Hier nun die Regeln und Angebote: (soweit ausgearbeitet)

1. Keine Mitgliedergebühren, keine Aufnahmegebühr, Anmeldebogen.
2. Info erscheint ca. 6-wöchentlich zu Selbstkosten.
Kein Mitglied ist verpflichtet die Info's zu beziehen!
3. Schüler, Studenten, Rentner und Arbeitslose erhalten die Infos (natürlich nicht rückwirkend die alten) zu 60% des Originalpreises nach Vorlage eines entsprechenden Nachweises. Dieser Nachweis ist natürlich **regelmäßig** zu erbringen, sonst verfällt die Begünstigung. Dies können wir durch die Reduktion der Druckkosten durch die vermutliche Mitgliederzahl von über 400 schaffen. Wer diese Vergünstigung ab dem kommenden Info haben möchte muß mir den entsprechenden Nachweis bis zum 24. Oktober 1985 zusenden!
4. Programme (Selbstgeschriebene) können weiterhin über den Club verkauft werden. Das sollte sich allerdings auf wirklich gute Programme beschränken! Einfache Programme und Public-Domain (d.h. Copyrightlose CP/M-Software) wollen wir mittels Wanderdisketten/cassetten weitergeben:

Zusammen mit Thomas Pflaum und auch mittlerweile den anderen Clubleitern habe ich vor Software auf eine möglichst preiswerte Art und Weise weiterzugeben. Dabei ist folgender Ablauf geplant: Jeder Diskette/Cassette wird eine Namensliste beigelegt, und jeder ist angehalten die Programme rasch zu kopieren und die Originaldiskette weiterzuschicken, und an denjenigen, der die Wanderdatenträger verwaltet, eine Postkarte zu schicken damit wir wissen, wo die Dinger sind. Sicherheitshalber wird auf die Disketten und Cassetten je ein BRIEF geschrieben, der dies noch einmal verdeutlicht und die Namensliste enthält.

Für diese Wandersoftware wird erst im nächsten Info eine Liste mit Hinweis, bei wem man sie bestellen kann, vorliegen. Erst dann wird es sie geben. Dabei werden die Disketten mit Programmen von Mitgliedern DM 1.- und solche mit Public Domain DM 5.- kosten. Das so eingenommene Geld soll die Disketten und den Kauf der Public Domain ermöglichen.

Funktionsstasten im BASIC (Herbert Oppmann)

Ja, es geht - und sogar recht einfach. Hier ein kleines Listing, wie man/frau es machen kann. Es stammt natürlich von einem BASIC-Experten: Herbert Oppmann. Von Herbert habe ich auch noch ein paar andere Kleinigkeiten (wie ein RENUMBER), die Ihr natürlich bekommen könnt. Ich will allerdings das RENUMBER so ändern, daß es NEW-Fest ist, und mit extended Basic zusammen laufen kann.

Um das Listing auf eine Seite zu packen habe ich es zweispaltig, ohne die Assembler-Zeilennummern geschrieben! Wichtig ist Herbert's Trick die ganzen Sprünge mit JR (jump relativ, relativer Sprung) zu schreiben, um es so verschieben zu können. Ich habe von zwei Labels die Adressen unten angegeben. Das Programmlisting ist folgendermaßen: Spalte links, Spalte rechts und dann unten den Rest.

Die Definition einer Funktionstaste geht nach der Syntax: Hintereinander für die F-Tasten die zugehörigen Strings, wobei im letzten Byte jeder F-Taste das 8. Bit gesetzt werden muß. D.h. RUN mit Return: "RUN",#8D. Es müssen alle Funktionstasten definiert werden!

In der Zeile TABELL: LD HL,#F039 wird in das Register HL die Adresse der Tabelle TEXTE nach dem Verschieben geladen. Deshalb müßt Ihr das Programm genauso eingeben außer den Definitionen der Tasten in TEXTE. Wer (warum auch immer) den ersten Programmteil in der Länge ändert **muß** in der Zeile TABELL: ... die neue richtige Adresse an Stelle von #F039 einsetzen!

10 CODE

START:	LD HL,0	POP HL
	LD (#FABA),HL	RET
	LD HL,CUST	LESE: POP HL
	LD DE,#F000	CALL #5622
	LD BC,256	PUSH AF
	LD (#FD52),DE	CP 128
	LDIR	JR NC,TE
	RET	ERR: POP AF
CUST:	PUSH HL	RET
	LD HL,(#FABA)	TE: CP 144
	LD A,H	JR NC,ERR
	OR L	EX (SP),HL
	JR Z,LESE	TABELL: LD HL,#F039
ERST:	LD A,(HL)	SUB 128
	INC HL	AND A
	BIT 7,A	JK: JR Z,ERST
	JR Z,NOCH	WEI: BIT 7,(HL)
	LD HL,0	INC HL
	RES 7,A	JR Z,WEI
NOCH:	LD (#FABA),HL	DEC A
	AND A	JR JK

```

TEXTE: DB "DISC LOAD ",#A2,".BAS",#22,#8D,#A0,"DISC CTLC",#8D,
        DB "DISC SAVE ",#A2,"DISC DIR ",#22,"*.*",#22,#8D,#A0,
        DB "CLS:L.",#8D
        DB #A0,#A0,#A0,#A0,#A0,#A0,#A0,#A0
        RET

```

20 NEW

START auf Adresse 8007, TEXTE auf Adresse 8056

F D X B - M e n u (Herbert Herberg)

Ich stand vor dem Problem: Nun habe ich zwei Laufwerke, und wollte in einem FDXB haben und in den anderen die Programme (die evtl. auch einiges nachladen). Da bot sich die Idee eines Menü-Programmes an, das die Directory des C-Laufwerkes sortiert anzeigt, und das gewünschte Programm auf Tastendruck lädt. Als Regel haben die Hauptprogramme im Dateinamen keine Extension (d.h. keinen Punkt), und alles nachladbare hat eine EXT.

Das Programm wählt Laufwerk C an (Zeile 30), liest die Directory auf den VS 4 (ab 1000), wobei INK und PAPER gleich sind, da ich diese Dir noch nicht haben will. Dann wird die Directory mit SPK\$ auf das Feld DIR eingelesen und sortiert (1100). Nun zeige ich die Programmnamen auf dem 80-Zeichenschirm dreispaltig an (1300), wobei jeder Name eine Buchstaben (angefangen mit A) erhält. Ein Druck auf den Buchstaben lädt das Programm, ESC liest die Directory nochmals ein, damit ich die Diskette wechseln kann. Der Text in 1230 wird dabei oben auf dem Schirm zentriert in einem Kasten ausgegeben, und kann durch einen beliebigen anderen (auch anderer Länge, aber nicht viel länger) ersetzt werden. Abspeichern mit RUN.

Die Kommentare in *kursiv* gehören nicht ins Programm!

```

0 DISC SAVE "MENU"
10 DIM DIR$(64,8)
20 POKE 64862,13           Keinen BREAK zulassen
30 CODE
      LD C,14              BDOS: Drive Select
      LD E,2               BDOS: Laufwerk C (B=1, D=3)
      CALL 5              BDOS Aufrufen
      RET

1000 REM Directory einlesen
1010 VS 4: PAPER 5: INK 5: CLS : DISC DIR "*": CSR 0,0
1020 FOR ANZ=1 TO 60
1030 LET A$=""
1040 FOR I=1 TO 8: LET A$=A$+SPK$: NEXT Namen einlesen
1050 FOR I=1 TO 8: LET B$=SPK$: NEXT Extension übergehen
1060 LET DIR$(ANZ)=A$ Eintragen in DIR
1070 IF A$<>" " THEN NEXT ELSE LET ANZ=ANZ-1
      Ende wenn keine Datei mehr da

1080 INK 1: CLS : VS 5
1090 IF ANZ=1 THEN GOTO 1200 Einen Namen nicht sortieren!
1100 REM Directory sortieren Nicht schnell, aber klappt
1110 FOR I=2 TO ANZ
1120 FOR J=1 TO I
1130 IF DIR$(J)>DIR$(I) THEN LET H$=DIR$(I): LET DIR$(I)=DIR$(J):
      LET DIR$(J)=H$
1140 NEXT : NEXT
1200 REM Kopf
      X1, Y1 ist die obere linke Ecke des Kastens. Diese
      Werte dürfen verändert werden, aber unnötig!
      X1+1 und Y1+1 mu~ durch 4 teilbar sein!!
1210 LET X1=11: LET Y1=71: LET Y2=Y1+13: LET X2=159-X1: CLS
1220 LINE X1,Y1,X1,Y2: LINE X1,Y2,X2,Y2: LINE X2,Y2,X2,Y1:
      LINE X2,Y1,X1,Y1 Kasten

```

F D X B - M e n u / D I S C R E N & S A V E

```

1230 LET TEXT$="Memotech MTX - FDX Spiele Herbert Herberg"
      Die Schrift wird im Kasten zentriert
1240 CSR (X2+X1-2*LEN(TEXT$)+1)/4,25-Y2/4: PRINT TEXT$
1300 REM Menu
1310 LET L=INT(ANZ/3+.67)
      Spalte 1 bei X1, Spalte 2 bei X2 und Spalte 3 bei X3
1320 LET X1=10: LET X2=34: LET X3=58: LET Y1=8
1330 FOR I=1 TO ANZ
1340 IF I<=L THEN LET X=X1: LET Y=Y1+I ELSE IF I<=L+L THEN LET X=X2:
      LET Y=Y1+I-L ELSE LET X=X3: LET Y=Y1+I-L-L Welche Spalte ?
1350 CSR X,Y: PRINT CHR$(I+64);" ";DIR$(I);
1360 NEXT
1370 LET Y=Y+3
1380 CSR X1,Y: PRINT "ESC Redo" Und ESC für Nocheinmal
1400 REM Dialog
1410 LET A$=INKEY$: IF A$="" THEN GOTO 1410 ELSE LET I=ASC(A$)-64
1420 IF I>32 THEN LET I=I-32 Kleinbuchstaben -> Großbuchstaben
1430 IF I>0 AND I<=ANZ THEN CSR 28,7: PRINT DIR$(I);" wird geladen !":
      DISC LOAD DIR$(I) Anzeigen und Laden
1440 IF A$=CHR$(27) THEN GOTO 1000 ELSE GOTO 1410

```

Dieses Programm hätte man an sehr vielen Stellen mit dem XBASIC wesentlich schneller und übersichtlicher programmieren können, aber wer hat das schon, und außerdem sollte ein Menü auch ohne solchen Luxus laufen, damit der MTX nicht allzuviel unnötiges laden muß. Das Sortieren braucht natürlich eine kleine Weile!

DISC REN ist tückisch!

Unter CP/M liefert der REN (d.h. REName = Umbenennen), wobei eine Datei in eine existente umbenannt werden soll wie zu erwarten eine Fehlermeldung. Aber unser FDXB ist da nicht so pingelig. Warum sollte der selbe Dateiname auch nicht gleich doppelt vergeben werden. Hatten wir denn nicht dank Memotech (dSM) schon genug Ärger mit den kleingeschriebenen Dateinamen, da CP/M nur solche mit Großbuchstaben ansprechen kann. Aber nein, wenn schon denn schon. Also beim Abspeichern eines Programmes, dessen alte Version erst gesichert werden soll, müsst Ihr also folgendes eingeben:

```

DISC ERA "PROG.BAK"
DISC REN "PROG.BAK"="PROG"
DISC SAVE "PROG"

```

DISC SAVE ist nicht besser (Herbert Herberg)

Jedes halbwegs anständige Programm sollte eigentlich beim Versuch auf die Diskette zu schreiben bei zu wenig Platz auf der Diskette eine entsprechende Meldung ausgeben. Memotech war anscheinend der Meinung, daß etwas Nervenkitzel nicht schaden könnte (dSM), und FDXB speichert einfach solange Platz ist und hört dann einfach kommentarlos auf. Warum fragt Ihr Euch sicherlich. Wenn das Programm schon einmal auf der Diskette ist, dann sollte doch das DISC SAVE schaffen, indem das Original gelöscht wird - aber FDXB benennt die auf der Diskette gespeicherte Version um, speichert, und löscht dann die alte Version. Als Abhilfe kann man eigentlich nur zuerst das Programm mit DISC ERA beseitigen und dann DISC-SAVEen.

B A S I C Maskiertes Plotten / D. Gröning zu BASIC**Maskiertes Plotten** (Michael Köster)

Im letzten Info tauchte die Frage nach einem Verfahren auf, zu "plot-ten" und gleichzeitig Text zu betrachten .

Nun ,die Lösung ist ganz **einfach**: (Anm.d.Red.: "HAHA")

Bei der Ausführung der VS-Befehle ändert der Basicinterpreter die Register des VDP , sowie die entsprechenden Systemvariablen , die ihm anzeigen , in welchem Zustand sich der VDP befindet. Um zu verhindern das beim Ausführen eines Grafikbefehls im Textmodus eine Fehlermeldung das Programm abbricht , muß den Routinen der Grafikbefehle vorgegaukelt werden, das sich der VDP im Grafikmodus befindet ! Leider steht diese Systemvariable nicht im Handbuch . Ich habe sie aber trotzdem durch Vergleiche herausgefunden : Es ist die Adresse 65371 wobei im Textmodus hier der Wert 168 und im Grafikmodus 153 steht (fragt bitte nicht warum) .

```
Also:          10   CRVS 5,0,0,0,40,24,40
                20   VS 5
                30   PAPER 4 : INK 1
                40   PRINT "Hier ist Text"
                50   POKE 65371,153
                60   CIRCLE 128,96,95 : LINE 0,0,255,191
                70   POKE 65371,168
                80   CSR 0,5 : PRINT "Jetzt paß mal auf:"
                90   PAUSE 1000 : VS 4
               100   GOTO 100
```

Allerdings dürfte dann auch für Farbwechsel im Grafikmodus nur noch "poken" übrig bleiben.

Dietmar Gröning meint

Lieber Michael Möwe!

Da kauft man sich vor einem halben Jahr einen MTX rein zufällig, weil die Leute von VOBIS dieses Teil so günstig anpreisen (zwar im Nachhinein eigentlich nicht so, weil ich das zweite Laufwerk an Stelle des Druckers genommen habe), versucht sich ein wenig mit dem englischen Manual durch das Newword zu quälen, weil nämlich die Examensarbeit damit geschrieben werden soll, hat dann ein wenig Luft, die Feinheiten des Gerätes zu studieren, sucht verzweifelt nach Infos und findet nichts - bis auf die Adresse des Klubs. Aber dann schlägt alles ins Gegenteil um - ich habe kaum Zeit die monatlichen Mengen an Papier zu bewältigen, die da ins Haus kommen, weil ich ja auch nicht nur vor dem Apparat sitzen kann (siehe 8/23). Dann kommt also Michaels Anschluß - den ich verstehen kann, obwohl ich dieses Klagen nicht mag. Aber er hatte ja Erfolg.

Also: Infos sind Spitze, so manches was sich die Leute bei Vobis nicht träumen ließen und auch nicht vorführen konnten nämlich der Blocksatz (column) klärte Info 8.

RS 232 nach Bauanleitung läuft auch.

Was mit der Ram-Erweiterung sein wird, weiß ich erst wenn das Prom da ist-32K von 64k sind jedenfalls da.

Vor dem Info: Mtx eine Blackbox, die vielleicht doch nicht so gut ist , wie sie aussieht. Nach dem Info: Der mtX ist eine solide Sache, die noch viele Möglichkeiten der Bastelei enthält. (Was ist denn ein PC 10 oder gar IBM PC den man gar nicht mit dem LötKolben nahekommen darf ? Langweilig !)

D. Gröning zu BASIC

Aber nun zum Info 7: S 30/31 in dem netten und genial einfachen Z80 Assembler, der prima läuft sind zwei Fehler:

Im Teil code8200 und code8100 muß die Zeile 2 so lauten:

2 LET A= PEEK(N-1)*256+PEEK(N), da die Länge der CODE-Zeile in den ersten Beiden Bytes gespeichert sind; dann läuft die Sache ausgezeichnet, wenn an Zeile 10 zwei Zeichen angehängt werden.

Und nun zu MMs Tatendrang: Da die RAM-disk nicht mit dem FDXB zusammenläuft und ich ein kleines Dateiprogramm in fdxb habe, das Datensätze der Länge 128 (=1 Rec) =^= bei einer 40 Spurdisk 2496 recs speichern kann; und nach Stichworten durchsuchen kann, dafür aber ca. 5 Min braucht hatte ich auf die Ramdisk gehofft. Dieses Programm in Assembler wäre natürlich prima, weil dann ja schneller und dann mit der Ramdisk die man ja noch erweitern könnte, bzw. auf einem 80spur Laufwerk, da tun sich dann schöne Perspektiven auf. Aber leider werde ich nicht die Zeit haben, so etwas zumachen, weil ich das hier nur als Freizeitbeschäftigung betreibe.

Informationen zum Programm:

zuerst wird das Inhaltsverzeichnis der Disc gelesen und Dateien mit dem Zusatz .DAT aufgeführt. Natürlich ist es möglich auch auf Disc C: zu schreiben und zu lesen, wenn man lange Dateien hat. Im Programm ist dieser Teil einfach durch das REM stillgelegt.

Dann erscheint das Eröffnungsmenue und je nach Wahl geschieht dies oder das. Die einzelnen Programmteile sind klar getrennt, so daß die einzelnen Funktionen zu sehen sind. Daß die Auswahl der Bereiche am Ende steht hat nur den Grund, daß ich mit dem Auswahlverfahren angefangen habe.

Das akustische Signal chr\$(7) ist ganz praktisch, wenn man während der Suche in einer langen Datei irgendwo anders hinget. Die Uhr ist nur zu Kontrolle über die Dauer, sie kann ohne Einbußen an Fuktion einfach weggelassen werden.

Ich habe mit Büchern experimentiert, aber es ist möglich dieses Programm für jeden beliebigen Zweck umzuwandeln, durch umschreiben der Eingabe und Ausgabe Maske. Was nicht gelöst ist, ist eine Kombination von Suchbegriffen. Aber jeder kann daran basteln.

Nachteilig ist die lange Dauer der Bearbeitung und die Belastung des Laufwerks.

Das soweit.

Dietmar Gröning

Zu den Spielen: Willie ist einfach ein Renner, Missile ist langweilig, LABYRINTH ist sehr spannend, Lunar bringt einen Raumfahrer einfach zur Verzweiflung, weil man sich zu oft in den Boden bohrt. Die Grafik ist allerdings prima. Das wärs.

HEX-Zahlen in TURBO-PASCAL**HEX-Zahlen-Darstellung in TURBO** (Olaf Krumnow)

Wie jeder TURBO-User (und diese Gemeinde ist ja schon beachtlich) von den (zu oft auftretenden) Fehlermeldungen weiß, kann der Compiler Hex-Zahlen darstellen. Nur leider gibt es keine Funktion, um dieses auch in eigenen Programmen zu bewerkstelligen. Abhilfe schaffen folgende zwei Funktionen, die die entsprechenden Routinen im Laufzeitpaket des Compilers aufrufen. Dabei ist zu beachten, daß der Übergabeteil den Parameter **N:Integer** immer **als erstes** enthalten muß, weil in dieser Form der Wert im richtigen Register bereitgestellt wird.

Es wurden deshalb Funktionen gewählt, damit der Aufruf in einer write-Anweisung stehen kann, die ja bekanntlich nur Funktionen enthalten darf. Das Ganze kann auch leicht als Prozedur formuliert werden, es muß dann lediglich die Funktionszuweisung entfallen.

Eine Funktion muß auch eine Funktionszuweisung enthalten, deren Wert an die aufrufende Stelle übergeben wird, in unserem Fall also die write-Anweisung. Dies kann man ausnutzen, um der Hexzahl ein Zeichen nachzustellen, z.B. ein 'H' zur Verdeutlichung des Zahlensystems. Bleibt das Zeichen immer gleich, so kann es als Zeichenkonstante zugewiesen werden, wie unten auch geschehen. Für variable Darstellungen muß es im Aufruf übergeben werden. Das darf aber nur **hinter** dem Zahl-Parameter geschehen. Siehe dazu Funktion W_Hex2.

```
function W_Hex ( N : Integer ):char; (* N als 4-stellige Hexzahl *)
begin
  inline ($cd/$04af); (* Fuer TURBO 2.0 : Inline($cd/$048d); *)
  W_Hex := 'H'
end;
```

```
function B_Hex ( N : Byte ):char; (* N als 2-stellige Hexzahl *)
begin
  inline ($cd/$04b3); (* Fuer TURBO 2.0 : Inline($cd/$0491); *)
  B_Hex := 'H' (* siehe oben *)
end;
```

Mit I := 1 liefert der Aufruf write(W_Hex(i)) --> 0001H
und der Aufruf write(B_Hex(i)) --> 01H .

```
function W_Hex2 (N:Integer; C:char):char;
(* N als 4-stellige Hexzahl mit nachfolgendem beliebigen Zeichen *)
begin
  inline ($cd/$04af);
  W_Hex2 := C
end;
```

Jetzt liefert der Aufruf write(W_Hex2(257,'H')) --> 0101H
und der Aufruf write(W_Hex2(0,' ')) --> 0000

T u r b o - P a s c a l

Die Modulo-Funktion (Kurt-Bernd Rohloff)

Nun bin ich auch endlich dazu gekommen, die Infos aufzuarbeiten. Bei Info 7 angekommen, fiel mir auf Seite 10 eine Ungereimtheit auf. Die 5. Macke von Turbo Pascal ist m.E. genau andersherum:

-11 MOD 10 = -1

ist falsch, richtig wäre 9, während

-5 MOD 10 = 5

richtig ist. Warum?

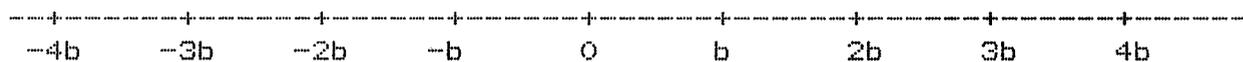
(Anm.d.Dipl.-Math.: negativ mod positiv sollte m.E. negativ sein!)

Was macht die Modulo Funktion? Die gängige Erklärung

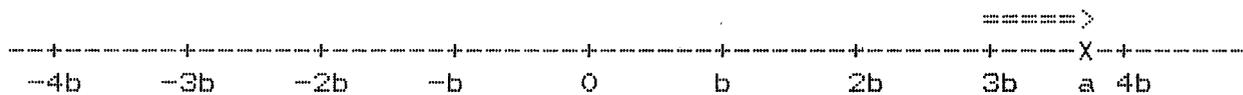
$a \text{ mod } b = a - (a \text{ DIV } b) * b$

ist nur für positive a gültig. Für negative a ist folgende Veranschaulichung günstiger:

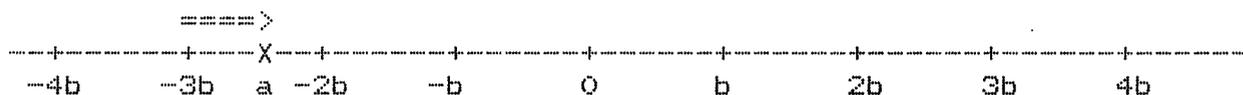
Stelle Dir bitte die Zahlengerade, beginnend bei 0, in Intervalle der Länge b eingeteilt vor:



Irgendwo auf dieser Zahlengeraden liegt nun a. Ich nehme zuerst einen positiven Wert an:



Dann ist a mod b gerade der Abstand von der nächst kleineren Intervallgrenze (0, wenn a genau auf einer solchen liegt). Genauso ist es bei einem negativem a:



In jedem Fall muß a mod b zwischen 0 (einschließlich) und b (ausschließlich) fallen. Frag mich jetzt aber bitte nicht, woher ich das weiß, das weiß ich nicht mehr, aber ich weiß, daß es so ist und daß ich es weiß, weißt Du?!

Leider verarbeitet der MOD Operator in Pascal nur INTEGER Werte, obwohl diese Einschränkung eigentlich nicht besteht. Was tun? Natürlich selbst ein Unterprogramm schreiben! Die folgende Funktion verarbeitet REAL Werte nach dem oben beschriebenen Verfahren, wobei -1 als Ergebnis geliefert wird, wenn b<=0 ist.

(Folgendes Programm ist im Prinzip von Kurt-Bernd, aber ich war so frei es etwas umzuschreiben, ohne es vom Ergebnis zu verändern, damit es übersichtlicher ist. Herbert)

Turbo - Pascal

```

FUNCTION Modulo(a,b: REAL): REAL;
  VAR z: REAL;
  BEGIN
    IF b <= 0
    THEN Modulo := -1
    ELSE BEGIN
      z := a;
      IF z >= 0
      THEN WHILE z >= b DO z := z - b
      ELSE WHILE z < 0 DO z := z + b;
      Modulo := z;
    END;
  END;
END;

```

Wo ich schon mal dabei bin, hier noch eine nette Umwandlungsroutine von Zahlen in Strings für Turbo-Pascal.

Zahlen in Strings umwandeln (Kurt-Bernd Rohloff)

Natürlich kenne ich die Str-Prozedur. Aber diese verwendet leider stets das Fließkomma-Format (auch wissenschaftliche Notation genannt), es sei denn, man gibt die Anzahl der Dezimalstellen mit an in der Form Str(x:laenge:dezimalen,Zeichen). Nun ist aber diese Anzahl häufig nicht im voraus bekannt. Hier schafft die folgende FUNCTION Str2 Abhilfe. Sie wandelt eine Zahl in einen String im Festkommaformat ohne führende Leerstellen, wann immer dies möglich ist. Sie wird wie folgt aufgerufen:

```
Zeichen:=Str2(x,slen)
```

wobei Zeichen als String der Länge slen vereinbart sein muß. Im aufrufenden Programm werden außerdem folgende Deklarationen benötigt:

```

CONST    slen=18; (* oder kleiner *)
TYPE     Zahlenstring=STRING(.slen.);
VAR      Zeichen:Zahlenstring;

```

ACHTUNG: Diese FUNCTION darf nicht in einer WRITE/WRITELN Prozedur aufgerufen werden!

```

FUNCTION Str2(x: REAL; len: INTEGER): Zahlenstring;
(* wandelt x in eine Zeichenkette im Festkommaformat, sofern moeglich.
Im Hauptprogramm muss man da zu vereinbart haben:
CONST slen = ...
TYPE  Zahlenstring = STRING[slen]
wobei Zahlenstring zwingend ist. Mit slen=18 klappt es immer.
slen darf aber auch kleiner sein. Bei groesseren Werten wird
automatisch auf 18 verkuerzt. Aufruf:
Zeichenkette:=Str2(Zahl,slen) *)

```

Turbo - Pascal

```

VAR      i                : INTEGER;
        rawdec,decim      : INTEGER;
        expo,ErrCode      : INTEGER;
        (* ErrCode ist nur fuer VAL noetig *)
        VorKomma,Stellen  : INTEGER;
        c                 : CHAR;
        Strx              : STRING[18];
        expostr           : STRING[3];
        Stand             : BOOLEAN;

BEGIN
Stand:=FALSE;
STR(x,Strx);                (* wandle |x| in String Strx *)
i:=14; c:=Strx[i];         (* 14=letzte Ziffer im String *)
WHILE c<>'0' DO           (* erstes Zeichen <>'0' suchen *)
    BEGIN                (* Pointer ist i *)
        i:=i-1; c:=Strx[i];
    END;
rawdec:=i-4;
expostr:=COPY(Strx,16,3);
IF expostr[1]='+'
    THEN DELETE(expostr,1,1); (* '+' entfernen *)
VAL(expostr,expo,ErrCode);  (* Exponententeil in *)
decim:=rawdec-expo;        (* eine Zahl umwandeln *)
IF ErrCode>0
    THEN WRITELN('VAL Fehler Nr. ',ErrCode,
        ' bei Zeichen ',expostr[ErrCode]);
IF decim<0
    THEN decim:=0;
VorKomma:=expo+1;         (* Vorkommastellen *)
IF VorKomma<1
    THEN VorKomma:=1;
Stellen:=2 + VorKomma + decim;
IF len > 18
    THEN len:=18;
IF Stellen>len
    THEN Stand:=TRUE;
(* WRITELN(VorKomma,'+',decim,'+2=',Stellen,', Stand=',Stand); *)
IF Stand
    THEN STR(x:len,Strx)
    ELSE BEGIN
        STR(x:len:decim,Strx);
        i:=0;
        REPEAT
            i:=i+1;
        UNTIL Strx[i]<>' ';
        DELETE(Strx,1,i-1); (* fuehrende Leerzeichen *)
        END;                (* entfernen *)
Str2:=Strx;
END; (*Str2*)

```

So, genug für heute. Nächstes Mal wahrscheinlich mehr.

TURBO-Editor / Assembler / Angebote / Drucker**Verwendung des guten TURBO-Editors** (Jürgen Marquardt)

Da haben wir nun einen schönen kleinen Editor, und wollen ihn anders verwenden: z.B. für dBASE-Kommandodateien, die mit .CMD enden:

Patching für TURBO 2.0 auf dBASE .CMD-Dateien:

(Eingaben **fett**, Kommentare *kursiv*, RET = /)

```
A>DDT TURBO.COM/                               >S2129/
>S 2C4C                                         2129 CD - 00/   Ohne Error-
2C4C 50 - 43/   ergibt .CMD                    212A 42 - 00/   Messages
2C4D 21 - 21/   als Extension                   212B 2C - 00/
2C4E 41 - 4D/   statt .PAS                      212C 32 - .
2C4F 53 - 44/                                       >^C
2450 32 - .                                         A>SAVE 119 EDITCMD.COM
```

422 Assembler-Befehle gratis ! (Hagen Wenzek & Bernd Preusing)

Vielleicht hat der eine oder andere schon mal davon gehört:

Die Z80 hat min.422 weitere Befehle mehr !!!!

Die sind zur Beeinflußung der IX und IY Register. Da (fast?) alle Assembler diese Befehle nicht verstehen, muß man die halt überlisten, indem man das ganze in DBs bzw. DEFBs unterbringt.

Irgendwo in diesem Info müßten die Befehle auch kopiert zu finden sein, also suchen !!

Wer es jetzt ausprobiert hat und sich wundert, daß nichts geht, der hat entweder irgendwas falsch gemacht, oder muß sich eine andere Z80 besorgen. Es geht nämlich nicht mit allen! Bei mir z.B. geht es !!

Grafiken (Frank Bueschler)

Ich suche noch sehenswerte Grafiken für eine Video-Show einer rollenden Diskothek. Sollten möglichst "nichts" kosten -> über Tausch läßt sich reden.

Adressetiketten / Papier (Frank Bueschler)

Etiketten, 89 x 36 mm, 1000 Stück DM 12.75 -siehe Info-Umschlag

Endlospapier, DIN A 4, 2000 Stück DM 45.00

Einzelblatt, DIN A 4 ,

(bestes Klopierpapier) 500 Stück DM 14.00

Das ganze natürlich wieder plus Porto und Verpackung, incl. MwSt.

Olympia-Drucker (Hagen Wenzek)

Ich habe herausgefunden, daß mein MS-15 ein Nachbau des Olympia "so-wieso" ist. Da es bei uns in Bonn einen Laden gibt, der für Olympia-Drucker das Eprom ändern kann, kann er das auch bei meinem Drucker.

Ich kann die EPROMs ohne weiteres bearbeiten! (Herbert Herberg)

Aufrüstung DMX80 für proportional und NLQ (Frank Bueschler)

Wie sicherlich inzwischen einige Leute wissen, ist der DMX80 "baugleich" mit dem Panasonic KXP-1090. Leider ist baugleich nicht baugleich, da Memotech diese Panasonic-Geräte auf dem "grauen" Markt kauft und teilweise die Technik ein wenig ändert. (Schon wieder 'ne dunkle Stunde von Memotech => **DSM**)

Das ist also das erste Problem, das zweite folgt auch sogleich: Eine Aufrüstung des KXP-1090 auf KXP-1091 ist laut Panasonic unmöglich. Somit besteht also keine Möglichkeit unseren Drucker zu erweitern.

H A R D W A R E**Aufrüstung MTX500 auf MTX512** (Frank Bueschler)

Der einzige Unterschied zwischen diesen beiden Geräten liegt in einem kleinen PAL und acht Speicher-IC's. Der 512er bringt es auf der Grundplatte auf 64kB, d.h. das man die 32kB-Erweiterung des 500ers voll als RamDisc nutzen kann. Auch wenn andere Speichererweiterungen ins Haus stehen, können 32kB mehr in der RamDisc schon nutzvoll sein. Gegen einen Aufpreis von DM 10,- kann bei Herbert Herberg das PROM für die 32kB-Karte getauscht werden.

Farbmonitore (Frank Bueschler)

Bei den bestellten Farbmonitoren gibt es leider derzeit einige Probleme. Da die Firma sich nicht gerade in der Nähe befindet, kann ich nicht genügend "Druck verbreiten". Allerdings hoffe ich, das ich in den nächsten Tagen auch dieses Problem gelöst haben werde.

VRAM-Controller und RESET (Hagen Wenzek)

Zuerst mal was für die Nichtfloppyuser bzw.MTX.COMer:

Den meisten, die mal auf die Platine geguckt haben, wird wahrscheinlich dieser kleine Steckverbinder zwischen den ROMs und dem TMS9929A aufgefallen sein.

Wie man leicht aus den Schaltplänen entnehmen kann, verbindet dieser den RESET Eingang zum TMS. Zieht man nun den Stecker raus und gibt einen RESET, bekommt der TMS nichts mehr davon mit und die Register im TMS müßten so sein wie früher. Leider hatte ich bis jetzt noch keine Zeit, das auszuprobieren. Wer das schon getan hat, oder es noch tut, veröffentliche dieses doch im nächsten Info.

MTX schneller ????? (Hagen Wenzek)

Wenn man nur den reinen MTX hat (ohne RS232-Karte!), kann man durch bloßes Austauschen des Quarzes von 4 auf 5 Mhz, den MTX um 1/4 schneller machen. Will man noch schneller werden, braucht man eine ZBOB CPU und Glück, daß man EPROMS, RAMs usw. erwischt hat, die mitspielen, sowie ein stärkeres Netzteil. Dieses brauchen FDXer auch schon bei 5Mhz. Leider weiß ich noch nicht, wieviel das leisten muß. Ausprobieren !!

Dazu meint Dipl.-Math. Herbert Herberg

Das große Problem beim Beschleunigen ist der Strom. Schneller kostet reichlich Saft. Außerdem müssen die (E)PROMS im MTX und auf der Buskarte der FDX gegen schnellere ausgewechselt werden. 5Mhz hat Frank Dersewski ohne FDX problemlos geschafft, mit FDX ist auf der 80-Z-Karte anfangs Mist, aber es klappt. 6Mhz klappte nicht. Übrigens kann es nicht schaden, den DART durch einen ZBOB DART zu ersetzen.

TEAC umschaltbares Laufwerk (Hagen Wenzek)

Mit meinem neuen TEAC G/F bin ich bis jetzt sehr zufrieden, wenn man mal von 2 Punkten absieht:

1. Die Elektronik ist auf der Unterseite von dem Ding und somit muß man erstmal die ganze dicke große schwarze Kiste auseinandernehmen, um an die Steckverbinder zu kommen.
2. Kaum ein Laden in ganz Deutschland hat scheinbar wirklich Ahnung: Ich habe bis jetzt erst 2(!) gefunden, die auf Anhieb wußten, was HD-Disketten sind. Und die wollten sage und schreibe Zweihundertundzehn Deutsche Mark haben (=DM 210,-). Muß man sich mal vorstellen. Die HD-Disketten aus dem letzten Info, sind auch so ein Fall von Unwissenheit.Es waren nur DD/DS 96TPI Disks.

Textmodus des Videocontrollers**Der Textmodus des TMS9929A (Michel Köster)**

Verwendung: Mit dem Betrieb einer 80Zeichenkarte scheint die Nutzung des 40Zeichenmodus fraglich. Hier dürfte das VRAM des VDP als 16k zum textzwischen speichern besser genutzt sein. Auch beim programmieren von Spielen, oder "Grafik"-Programmen gibt der Grafik1-Modus durch seine Mehrfarbdarstellung trotz 32-Zeichen wesentlich ansehnlichere Bilder bei gleichem Speicherbedarf. Erst die Notwendigkeit Zeichen definieren zu können und den Verzicht auf Farbdarstellung zu Gunsten einer, optischen Anpassung an den 80Zeichenmonitor rechtfertigen den Einsatz.

Die direkte Programmierung: Unter CP/M sind beim VDP alle Register auf Null gesetzt, nur unser MTX-ROM und das FDXBasic initialisieren diesen Baustein. Werden nun die Werte 208 und 129 in dieser Reihenfolge vom Z80 in den Port2 geschrieben befindet sich der VDP anschließend im Textmodus. Dies geschieht durch die entsprechenden Befehle einer höheren Sprache, oder in Maschinensprache. Wobei auch im ersten Fall diese Aufgabe durch "OUT"-Befehle erledigt wird. Nun muß der Benutzer erst einmal entscheiden welche Farben er im Text verwenden will. Wollen wir uns optisch nach dem 80Zeichen-Bild richten käme eine Ausgabe der Werte 32 und 135 in Frage (32=20H=Vordergrund:Grün(2)-Hintergrund:Schwarz(0)/135=87H= 8:"Registerschreibmodusbit"-7:Registernummer). Der Bildschirm bleibt weiterhin Schwarz, es fehlen noch die Daten für den Bildschirmspeicher und den Mustergenerator. Zunächst müssen wir den Speicherbereich wählen in dem diese beiden Tabellen geschrieben werden sollen. Hierfür sind die Register 2 und 4 vorgesehen. Register 2 enthält die Anfangsadresse des Bildspeichers, genauer gesagt seine oberen 4 Bits. Hier kann also in 1k-Schritten "gepaged" werden ($2^4=16$: $16/16=1$). Register 4 enthält die oberen 3 Bits der Anfangsadresse des Mustergenerators. Das bedeutet die Auswahl der Zeichensätze in 2k-Schritten ($2^3=8$: $16/8=2$). Liessen wir nun beide Register im Leerzustand, lägen Bildspeicher und auch Mustergenerator ab der 0000H-Adresse. Dagegen ist im Prinzip nichts einzuwenden, wenn einige Regeln beachtet werden. Der Bildspeicher ist im Textmodus 960 Bytes lang, je 1 Byte vertritt eine Cursorposition des Videobildes. Diese Bytes "zeigen" auf 8 Byte des Mustergenerators, funktionieren also als untere Bits des Registers 4. Findet der VDP z. B. an einer Stelle den Wert 15, multipliziert er ihn mit 8 und addiert mit dem 2048fachen Wert aus Register 4. In diesem speziellen Fall ($15*8+0*2048$) werden die Bytes der Adressen 120-127 als Muster interpretiert und an der entsprechenden Bildposition für den Monitor ausgegeben. Dabei ist zu beachten, daß der VDP im Textmodus die niedrigsten 2 Bits jedes Mustergeneratorbytes ignoriert, also nur 6 Punkte horizontal ausgibt !

An diesem Beispiel erkennt man bereits die Fehler, die auftreten können, wenn Bild- und Musterspeicher sich überschneiden.

Die Bytes 120-127, die ja im Beispiel ein Zeichen darstellen sollen, liegen im gleichzeitig Bildspeicher und zeigen ihrerseits auf weitere 8-Byte-Blöcke. Sicherlich kann man auch einige lustige Effekte mit dieser "Methode" erzielen, doch dazu gehört schon ein ausgeklügeltes System.

Textmodus des Videocontrollers

Unsere Register 2 und 4 stehen immer noch auf 0 und solange dürfen in den 960 Bytes, die vom Bildspeicher (0000H-03BFH) mitbenutzt werden, keine Werte stehen die mit 8 multipliziert eine Zahl kleiner als 960 ergeben. Nutze ich nun den restlichen Speicher meines 2k-Generators bleiben mir 136 Zeichen definierbar. Durch die Ausgabe von 1 und 130 auf Port2 schalte ich den Bildspeicher auf den Bereich von 0400H-07BFH um. Damit ergibt sich eine ähnliche Situation wie im vorhergehenden Fall, nur kann ich jetzt ASCII-Codierungen direkt in das VRAM schreiben (der ASCII-Code umfasst die Werte 32-127).

Mit der Ausgabe von 2 und 130 trenne ich Bild- und Mustergenerator und stehe jetzt vor einer Auswahl von 256 Zeichen. Mir bleiben dabei noch 14 unabhängige Bildseiten.

Wie bereits am Anfang erwähnt bietet sich der 40Zeichenmodus immer noch zur Zeichendefinition an. Anwenderprogramme in Chemie, Physik, Mathematik, Biologie uva. sollten auf solche Möglichkeiten nicht verzichten.

Tips und Tricks: Durch verändern eines einzigen Zeichens in der Mustertabelle, kann ich sämtliche Inhalte des Bildschirms, die auf dieses Zeichen zeigen verändern. Dadurch lassen sich einfach bestimmte Symbole, durch blinken o.ä., hervorheben.

Durch interruptgesteuertes Ändern des Registers 7, können mehr als 2 Farben gleichzeitig dargestellt werden. Werden 100 mal in der Sekunde die Werte 64 und 32 abwechselnd eingeschrieben, käme ein Bild zustande das zur Hälfte mit blauer, zur anderen Hälfte mit grüner Schrift gefüllt ist. Durch Ausgabe von 0 und 135 auf Port2 kann ich mein PAL-Bild abschalten.

Anhang: Zur direkten Programmierung werden in Assembler sämtliche OUT und IN-Befehle benutzt - FDXBasic bietet den OUT-Befehl und die INF-Funktion - MBasic benutzt ebenfalls OUT-Befehle und IN-Funktionen. Genaueres siehe unten.

Einige Befehle für den Textmodus:

Befehlsreihe (Port)	;Funktion
OUT 208 (2) ; OUT 128+1 (2)	;schaltet den Textmodus ein.
OUT X*16+Y(2);OUT 128+7 (2)	;wählt Farbkombination, wobei X die Zeichen- und Y die Hintergrundfarbe ist.
OUT X (2) :OUT 128+2 (2)	;wählt die Startadresse für den Bildspeicher (0<=x<=15) Adr.=X*1024
OUT X (2) :OUT 128+4 (2)	;wählt die Startadresse für den Zeichengenerator (0<=X<=7) Adr.=X*2048
OUT X (2) :OUT 64+Y (2)	;lädt den Read/Write-Counter mit einer Adresse und kündigt eine Writesequenz an. (0<=X<=255) (0<=Y<=63) Adr.=Y*256+X
OUT X (1)	;lädt den Wert X an den Punkt des VRAMs, der vom R/W-Counter adressiert wird. (0<=X<=255) RWC=RWC+1
OUT X (2) :OUT Y (2)	;lädt den RWC mit einer Adresse und kündigt eine Readsequenz an. (0<=X<=255) (0<=Y<=63)Ad. =Y*256+X
X= IN (1)	;holt das Byte des VRAMs das durch den RWC adr.wird (0<=X<=255) RWC=RWC+1

S I N U S / C - S p r a c h e**SINUS * anders** (Michael Köster)

Zum Schluss noch ein Einzeiler mit einer lustigen Verwendung der Sinusfunktion beim direkten Programmieren des Soundchips:

```
1 FOR X=0 TO 6. 28 STEP. 01:OUT(6), SIN(X)*255:A=INP(3):NEXT:GOTO 1
```

C (Olaf Krumnow)

Kürzlich las ich in einer Computer-Zeitschrift, daß in Nürnberg ein Club gegründet wurde mit dem Ziel, die Sprache C - jene geheimnisumwitterte "Alleskönner-Supersprache" - in Deutschland stärker zu verbreiten. Auf meine Anfrage erhielt ich nach wenigen Tagen sechs Seiten Text mit den Vorstellungen und (überwiegend noch) geplanten Aktivitäten des Clubs. Hier eine kurze Zusammenfassung:

- Herausgabe einer monatlichen Zeitschrift
- Verteilung und Anpassung von Public-Domain Software
- Herausgabe eigener PD-Software
- Erstellung einer Software-Bibliothek
- Auswertung von C-relevanten Artikeln
- Telefonseelsorge durch einen C-Profi
- Beratung in Hard- und Softwarefragen

Da nicht jeder einen C-Compiler besitzt und diese Dinger auch ganz schön teuer sind, erhält man nach Beitritt auf Wunsch einen Small-C-Compiler (englische Dokumentation, warum nicht eingedeutscht?) kostenlos. Der Club soll nach amerikanischem Vorbild professionell aufgezogen werden; wenn ich das richtig verstanden habe, sogar mit eingestellter Schreibkraft (Da kann ich nur sagen: hoffentlich. Der Text war wirklich grausam zu lesen). Um das zu finanzieren, beträgt der Jahresbeitrag DM 100.- Das klingt zunächst viel, aber rechnet mal nach: Bei 10-12 Infos im Jahr kommt das auch bei uns locker zusammen.

Für Interessenten hier noch die Anschrift des Clubs (Rückporto beilegen !):

C-Club
Vollmarstr.2
8500 Nürnberg 90

Ich bin diesem Club beigetreten (C reizte mich schon länger) und werde Euch über Ergebnisse, die uns allgemein interessieren könnten, auf dem laufenden halten, sowie natürlich für den Vertrieb der erscheinenden PD-Software sorgen.

H A R D W A R E / C I u b**Tastaturdekoder-Schalter** (Hagen Wenzek)

Noch was zur Hardware: Mit diesen kleinen blauen Dingern neben den Joystickports sind die auf Seite 179 im deutschen und Page 182 im englischen Handbuch gemeinten Schalter, zum Umschalten der Tastaturbelegung gemeint. Wenn man da rumprobiert bekommt man die englische, amerikanische, schwedische usw. Tastatur auf den Bildschirm. Den gleichen Effekt hat ESC B0, B1 usw., nur bleibt das jetzt immer. Wie gesagt, gilt das natürlich nur für MTX Betrieb.

Diese Schalter dienen genau dazu, den Tastaturdekoder auf deutsch bzw. englisch voreinzustellen. (Herbert Herberg)

Netzteil (Frank Bueschler)

Auch ich habe inzwischen mein Gehäuse der FDX in die Ecke gestellt und besitze nun ein IBM-Look Gehäuse, das sich nach oben aufklappen läßt. Hier habe ich nun ein neues Netzteil untergracht, daß in vielen Apple-kompatiblen Computern verwendet wird. Es besitzt keine Trafos, sondern erzeugt durch Transistoren und Widerstände eine absolut gleichmäßige Spannung ohne große Wärmeentwicklung. Preis : DM 100,-.

VOBIS neue MTX-Floppy: SDX (Frank Bueschler, Herbert Herberg)

Nach neuesten Information wird VOBIS weiterhin Memotech vertreiben, allerdings hat Memotech die FDX-Station ersetzt. Nicht etwa in Richtung PC, also endlich den Rechner mit in die FDX gesetzt, **nein**, genau umgekehrt. Die B0-Zeichen Karte wird bei den neuen Geräten auch im Rechner sitzen, was sich sicherlich nicht positiv auf die schon zu hohe Wärmeentwicklung auswirken wird. Die Laufwerke sollen 5 1/4 Zoll-Größe behalten.

Herbert Herberg hat erfahren: In die Tastatur kommt eine kombinierte B0-Zeichen/RS232-Karte ohne Platz für die notwendigen 32kB die der MTX 500 braucht. (Wir können allerdings abhelfen, s. Hardwareangebot auf den ersten Seiten). Der Floppycontroller wird links aufgesteckt und das B0-Spur-Laufwerk (Typ 07) kommt in ein schnuckeliges kleines Gehäuse. (Commodore 64 - like, meint Frank)

Geld schickt mir bitte auf eine der drei folgenden Arten:

1. Überweisung auf 34 8000-200
2. Scheck, gedeckt, EURO nicht notwendig
3. Bargeld, aber kein Einschreiben!

Ich mag keine Kopien von Einzahlungsbelegen!!!!!!!!!!!!!!!!!!!!

Kosten für Info9: **DM 10.80**

Einsendeschluß für Beiträge zum nächsten Info: 20. Oktober 1985

Erscheinungstermin des o.g. Infos: 1. November 1985

Clubtreffen des kontinentalen MTX User-Clubs Deutschland

Termin: Einlaß 13.00 Uhr am 16.11.85, Beginn 14.00 des selben Tages, Ende Nachmittags 17.11.85.
(Natürlich mit Pause in der Nacht!)

Ort: Hochschule für Wirtschaft und Politik (HWP) der Universität Hamburg, von Melle Park 9, 2000 Hamburg 13, im Gebäude ausgeschildert. Karte hierzu im nächsten Info.

Um die Übernachtung zum einen preiswerter zu gestalten, und zum anderen die Möglichkeiten des Kennenlernens zu verbessern will ich versuchen alle von Außerhalb Anreisenden bei Mitgliedern aus dem Raum Hamburg unterzubringen! Bitte meldet Euch also so frühzeitig wie es geht an, damit ich die Übernachtungen arrangieren kann; außerdem werde ich helfen Fahrgemeinschaften zu bilden. Aus diesen Gründen ist der **Anmeldeschluß der 1. November 1985.**

Für die Anmeldung ist der beiliegende Anmeldebogen zu verwenden, sowie eine an Euch adressierte und frankierte Postkarte beizulegen. Die Karten werde ich am 4. 11. 85 an Euch mit folgenden Informationen zurückschicken:

1. Bei Dir übernachten ... bzw. Du kannst bei ... übernachten.
2. Aus Deiner näheren Umgebung reisen ... an.

Zum Treffen bringt bitte alle ein paar Getränke und etwas zu knabbern mit. Abendbrot könnt Ihr ja mit Euren Gastgebern bzw. zu Hause essen. Die beiden bisherigen Treffen haben jedenfalls gezeigt, daß:

'Umso MTX/FDX desto weniger Essen aber reichlich Trinken' !
Eine Kaffeemaschine und Kaffee werde ich stellen!

Genug der Formalia!!!

Ich möchte gerne mehrere Anlagen auf dem Treffen stehen haben, damit alle möglichen Dinge vorgeführt werden können. Aber bitte bedenkt: **KEINE RAUKOPIEN** auf dem Treffen!!!

Auf dem Anmeldebogen könnt Ihr angeben, ob Ihr bereit seid eine Anlage mitzubringen. In dem Fall werde ich mich mit Euch direkt in Verbindung setzen, damit nicht zu viele sich die Mühe machen!

Das genaue Programm steht noch nicht fest, und wer sich bis zum 20. Oktober anmeldet, dessen Wünsche können leicht mit aufgenommen werden.

Folgende Programme/Anwendungen werden vorgeführt:

- NewWord, insbesondere NWINSTAL
- Rechnerkopplung mit MDM717 ... und evtl. CONTACT
- Grafik und Ton unter TURBO-PASCAL
- RAM-Disc mit SPOOLER
- Alle Programme, die aus der Bibliothek gewünscht werden
- Umschaltbare TEAC-Laufwerke
- PROM mit neuem 80-Zeichen-Zeichensatz
- Umbaumöglichkeiten für MTX/FDX
- Falls es klappt: ECB-Bus-Anschluß

Und natürlich einiges von dem, was Ihr vorführen wollt.

Hallo Herbert!

Mellrichstadt, den 12.08.85 24

1. C.Löhrmann schreibt in seinem Kompendium, DISC RÉN würde eine Fehlermeldung ausgeben, wenn eine Datei gleichen Namens bereits existiert. Zumindest mein Basic (272 recs) macht das aber nicht. Ich habe dann zwei gleichnamige Files im Directory. DISC LOAD lädt einen davon (welchen habe ich nicht untersucht). DISC ERA löscht beide. Suspekt, das ganze.

2. Im FDXBNEU von C.Löhrmann bzw. deinem FDXB+ ist ein Fehler. Er hat bei mir bewirkt, daß ein Basic-Programm verlustig ging, von dem ich kein Backup hatte, und das Listing war auch schon eine Arbeitsstunde alt. (Da zeigt sich wieder mal der Wert der Backups!)
Das Programm war zwar im Directory eingetragen, die angegebenen Blöcke auch beschrieben, aber mit Unsinn. Ladeversuche lieferten nur noch "No data". Da war nichts mehr zu retten. Meine Untersuchungen ergaben:
Wenn eine neue Diskette eingelegt wird und mit DISC CTLC angemeldet wird, kann zwar problemlos davon gelesen werden, DISC SAVE schreibt jedoch immer wieder den gleichen Sektor (£0080 bis £00FF) auf die Diskette, sooft wie das Programm Sektoren hat. Nach Auftreten eines "No data"-Fehlers funktioniert wieder alles richtig. Aber jedesmal so einen Fehler hervorzurufen ist natürlich keine Lösung.
Die Erleuchtung kam, als ich in einem CP/M-Buch las, daß Ctrl C die DMA-Adresse auf £0080 setzt. £0080 bis £00FF ist der Bereich, den CP/M als Sektor-Puffer benutzt. Im FDXB liegt dieser Bereich jedoch bei £7080 bis £70FF. Bei DISC CTLC muß also nach Funktion 13 =Ctrl C noch Funktion 26 = DMA-Adresse setzen ausgeführt werden damit alles klappt.

Also: 7810 0e LD C,13

```
11 0D
12 CD CALL £6308
13 08
14 63
15 11 LD DE,£7080
16 80
17 70
18 0E LD C,26
19 1A
1A C3 JP £6308
1B 08
1C 63
```

3. Wie bereits bekannt arbeitet die Basic-Funktion MOD bei MTX und FDXB nicht richtig. Hat schon mal jemand herausgefunden, woran das liegt und ev. eine Änderung für das FDXB gestrickt?

4. Ich habe am Wochenende zum erstenmal an die FDX einen Monitor angeschlossen. Große Enttäuschung! Die 80Z-Karte geht nicht. Habe FDX verpackt und meinem Cousin K.Mürding mitgegeben zu VOBIS. Bin gespannt, was daraus wird. Dadurch verzögert sich leider die Fertigstellung gewisser Programme, die ich dir versprochen habe, z.B. LIST.BAS :vollständig überarbeitet und an die Bedürfnisse von FDXB angepaßt.

-alle Eingaben können durch (Ret) mit angezeigten Default-Werten belegt werden

-Eingabe einer Org-Adresse. Z.B. Inhalt des Boot-PROMs, das ja nach Einschalten von £2100 nach £ec00 verschoben wird, kann von £2100 ausgelesen werden und angezeigt werden, als stünde er ab £ec00

-max 86 beliebige aufeinanderfolgende Sektoren können ab £9300 eingelesen werden und entweder nach breaken des Programms mit PANEL angesehen werden oder mit LIST.BAS disassembliert werden (interessant hierbei: Org-Adresse =£0100)

-läuft auf 80- und 40-Zeichen

Das wars vorerst wieder einmal. Tschüß

Herbert Oppmann

VS 4: Einzelpunkte (Klaus Muerling)

Manchen HI-Res Grafik-Designer wird interessieren, aus wieviel Einzelpunkten sein Kunstwerk denn nun besteht. Die nachfolgende Routine ermittelt dies. Das Ergebnis steht nach Aufruf in RESULT.

```

800F      LD DE,0
8012      CALL VSETRD
8015      LD C,24
8017 LOP1: LD B,0
8019 LOP:  CALL VDINPT
801C      DJNZ LOP
801E      DEC C
801F      JR NZ,LOP1
8021      LD (RESULT),DE
8025      RET
8026 VSETRD: PUSH AF
8027      LD A,E
8028      OUT (2),A
802A      LD A,D
802B      AND 63
802D      OUT (2),A
802F      POP AF
8030      RET
8031 VDINPT: PUSH AF
8032      IN A,(1)
8034      CALL TEST
8037      POP AF
8038      RET

8039 TEST: BIT 0,A
803B      CALL NZ,SUMM
803E      BIT 1,A
8040      CALL NZ,SUMM
8043      BIT 2,A
8045      CALL NZ,SUMM
8048      BIT 3,A
804A      CALL NZ,SUMM
804D      BIT 4,A
804F      CALL NZ,SUMM
8052      BIT 5,A
8054      CALL NZ,SUMM
8057      BIT 6,A
8059      CALL NZ,SUMM
805C      BIT 7,A
805E      CALL NZ,SUMM
8061      RET
8062 SUMM: INC DE
8063      RET
8064 RESULT: DS 2
8066      RET

Symbols:
VSETRD 8026      LOP1      8017
VDINPT 8031      LOP        8019
TEST    8039      SUMM       8062
RESULT 8064

```

Frank Bueschler

Kaffee- und Zigarettengehd
 Also man/frau wenn man sich das mal in die Kiste zieht,
 was Herbert und ich so verqualmen und versaufen bei so
 einer
INFO-REDAKTIONS-SITZUNG:
 Literweise Zigaretten und stangenweise Kaffee, ab und zu
 mal ein Fertiggericht aus der Dose und ein paar Streich-
 hölzer, die zwischen dem Augenlid und dem unteren Ende des
 Auges befestigt werden.
 Das Produktergebnis ist bei Beachtung des »ökonomischen«
Prinzips durchweg wirtschaftlich und *real* betrachtet auch
 effektiv, so jedenfalls die Meinung der **PRODUZENTEN**.
 Wir hoffen natürlich, daß Ihr mit uns auf dieses Ding
 namens Info abfahrt und auch im *Memory* habt, daß sich die
 Ausgabe von Kaffee- und Zigarettengehd jedesmal lohnt.
 So, das war also einer der berühmten Lückenfüller, denn
 den Nutzlücke hatten wir bei dieser Ausgabe ausnahms-
 weise mal nicht.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.

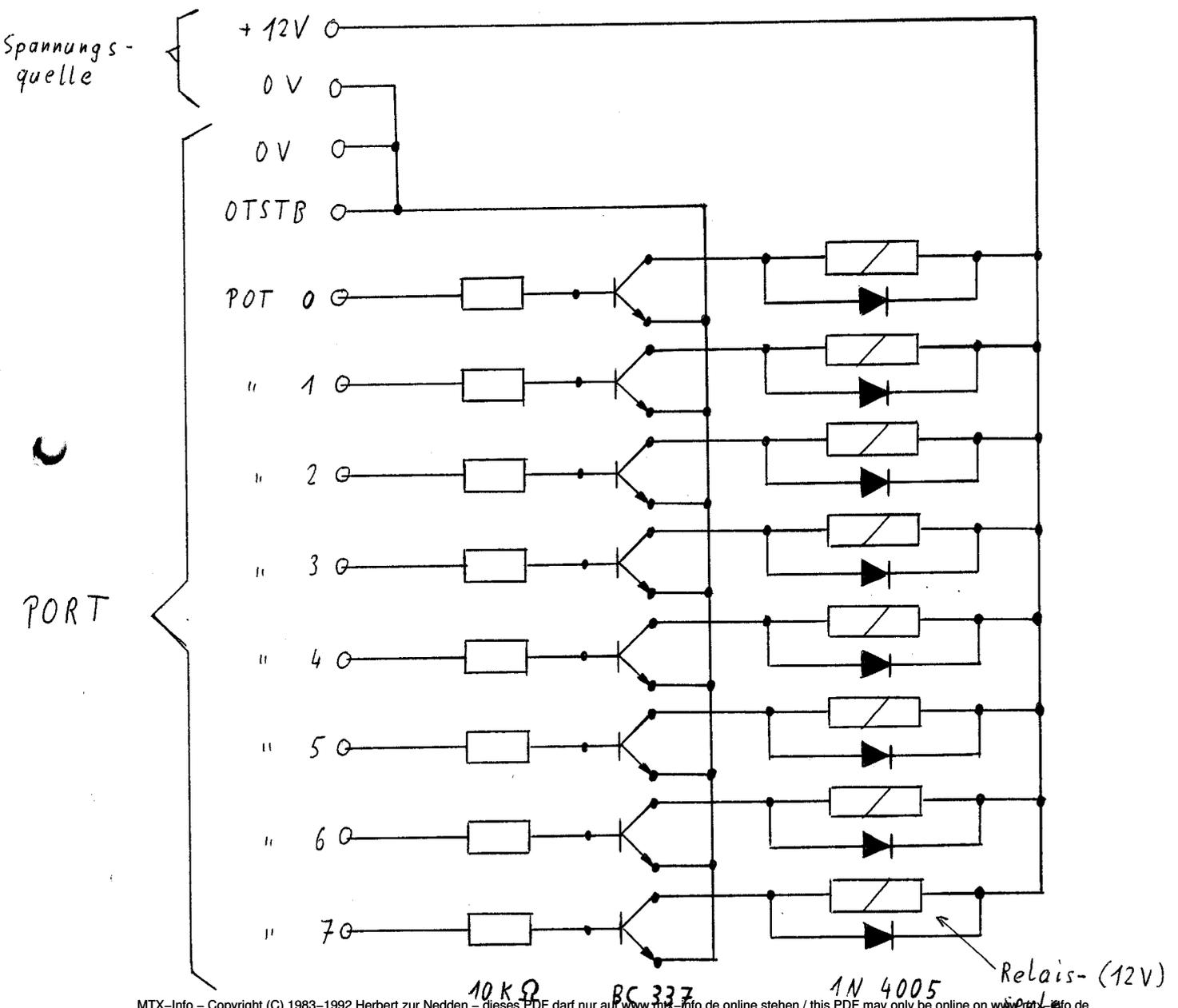
ANSTEUERUNG VON RELAIS MIT DEM M T X

Ich habe den parallelen I/O-Port (freier IC-Sockel neben der CPU) mit einem Stecker, wie er für die RS-232 Schnittstelle verwendet wird, an der linken Seite des MTX herausgeführt, um mir so ungestörten Zugang zu verschaffen. Diesen Port kann ich zum Beispiel zur Ansteuerung von Relais benutzen, so wie es unten dargestellt ist. Um Relais 1 anziehen zu lassen, gebe ich folgenden Befehl ein:

```

OUT (7),1 <RET>
Relais 2:   OUT (7),2 <RET>
Relais 3:   OUT (7),4 <RET>   usw.
    
```

Für FDX-Besitzer dürfte diese Hardwareerweiterung von Interesse sein, da sich hiermit zum Beispiel die beiden Monitorausgänge von MTX und FDX softwaremäßig umschalten lassen. Für weitere Experimente (Lauflicht, zeitabhängige Steuerungen, usw.) sind keine Grenzen gesetzt.



Füllstandsmessung im Soundbuffer

Der MTX besitzt die angenehme Fähigkeit, neben einem BASIC-Programm gleichzeitig Musik abspielen zu können mit Hilfe von SOUND und dem Tonpufferbefehl SBUF. Wenn man allerdings für längere Stücke nicht riesige Speichermengen durch SBUF verschenken will, um keinen Pufferüberlauf zu bekommen, tut man am besten nur einen Teil des Stückes in den Puffer und füllt ihn von Zeit zu Zeit nach, wenn er jeweils fast leer ist. Leider weiß ein BASIC-Programm nicht, wann der Puffer leer ist. Das Handbuch gibt auf Seite 123 Hinweise auf PEEKs, allerdings sind sie unvollständig und falsch übersetzt.

Daher werde ich hier einen Füllstandsmesser für SBUF ausführlich erläutern.

Ab Adresse 64082 sind jeweils 10 Byte Informationen über jeden der vier Soundbuffer reserviert.

PEEK(64082+KANAL*10+9) ergibt 1, wenn der Puffer nicht leer ist, sonst 0. Dabei geht KANAL von 0 bis 3.

Aber auch der direkte Füllstand läßt sich messen. Zu jedem Kanal gibt es zwei Zeiger. Wenn der Puffer leer ist, zeigen sie auf die gleiche Adresse. Sobald ein SOUND-Befehl kommt, wird dessen Information an die Stelle im Tonpuffer eingetragen, auf die der Schreibzeiger zeigt, anschließend wird er um 1 erhöht. Mit jedem SOUND wächst also der Schreibzeiger. Sobald ein Ton fertig gespielt ist, wird der Lesezeiger um 1 erhöht und die neue Toninformation gelesen. Dies geht solange, bis der Lesezeiger den Schreibzeiger eingeholt hat und der Puffer damit wieder leer ist. Wenn einer der Zeiger die Obergrenze des Puffers erreicht, fängt er einfach wieder unten bei 0 an, so daß sich in der Praxis ein ringförmiger Puffer ergibt. Falls der Schreibzeiger den Lesezeiger zu überholen versucht, kommt es zu Informationsverlusten.

Kurz und gut, den Füllstand mißt folgendes Unterprogramm:

```

100 LET MAXBUF=20: LET BASE=64082
110 SBUF MAXBUF
...
300 LET KANAL=0:GOSUB 5000
...
980 STOP
5000 REM Füllstands-Unterprogramm
5010 LET FUELLSTAND=PEEK(BASE+KANAL*10)-PEEK(BASE+KANAL*10+4)
5020 IF FUELLSTAND<0 THEN LET FUELLSTAND=FUELLSTAND+MAXBUF
5030 RETURN

```

In Zeile 5010 ergibt das erste PEEK den Schreibzeiger, das zweite PEEK den Lesezeiger, die Differenz ist der Füllstand. Es sei denn, der Schreibzeiger ist gerade von der Obergrenze des Puffers wieder an den Anfang gesprungen. Dann ist das Ergebnis negativ und man muß die Länge des Tonpuffers dazuaddieren.

Dabei ist es für alle diese Betrachtungen völlig uninteressant, wo der Tonpuffer tatsächlich im Speicher liegt und wieviele Byte Eintragungen dort pro Ton tatsächlich stehen. Als einzige Einschränkung bestehen die Zeiger nur aus einem Byte, so daß man für SBUF nie größere Argumente als 255 verwenden darf.

Praktische Anwendung

Am besten definiert man am Anfang des Programms z.B. mit SBUF 30 einen Puffer mittlerer Größe und füllt ihn dann mit

SOUND KAN, FREQ, LAUT, FREQINKR, LAUTINKR, DAUER, MODUS solange, bis der FUELLSTAND gleich 29 ist. Dann kann das Programm etwas anderes tun, und es muß nur von Zeit zu Zeit nachsehen, ob der FUELLSTAND unter 10 gesunken ist, und eventuell nachfüllen.

Jan Bredereke

```

10 CSR 9,0: PRINT "TERMINAL EMULATION"
20 CSR 13,3: PRINT "FOR MTX 512"
25 CSR 8,6: PRINT "-Type <ESC> to EXIT me": CSR 8,7: PRINT "Use Channel RS232"
27 FOR I=0 TO 38: PRINT "*": NEXT
30 CSR 0,12: INPUT "BAUD RATE: ";BR
40 BAUD 0,BR
45 PRINT CHR$(30): REM CURSOR ON
50 CLS
200 CODE

```

```

40D8 START: LD HL,START
40DB EI
40DC REC: IN A,(#E);CTL RS A
40DE BIT 0,A;DATA AVAILABLE?
40E0 JR Z,KEY
40E2 IN A,(#C);GET DATA RS A
40E4 OR A;STILL NO DATA
40E5 JR Z,KEY;JMP KEYB IF NOT
40E7 JR SCRIN;DISPL ON SCREEN
40E9 JR REC
40EB KEY: XOR A;CLEAR A
40EC CALL #79;KEYSCAN
40EF JR Z,REC
40F1 LD E,A
40F2 CP #1B;ESC ??
40F4 JP Z,END;YES, STOP
40F7 JR OUTPUT;OUT TO RS A
40F9 OUTPUT: IN A,(#E);CTL RS A
40FB BIT 2,A;LOOP UNTILL TRANSM
40FD JR Z,OUTPUT;BUFFER IS EMPTY
40FF LD A,E;LOAD DATA IN ACC
4100 OUT (#C),A;OUT TO RS A
4102 JR REC
4104 SCRIN: LD C,A
4105 LD B,0
4107 RST 10
4108 DB 192
4109 JR REC
410B END: RET
410C RET

```

Symbols:

```

START40D8REC40DC
KEY40EBSCRIN4104
OUTPUT40F9END410B

```

```

210 CSR 10,24: INPUT "***EXIT?(Y/N)**";S$
220 IF S$="N" THEN CLS : GOTO 10
230 IF S$="Y" THEN GOTO 240 ELSE GOTO 210
240 CLS : STOP

```

IM HINTERGRUND PLOTTEN

wollte der Grafik-Experte oder Manitu. Das ist wirklich nicht gerade einfach. Man muß nämlich aus der Plotposition die Adresse im VRAM berechnen, dort ein Byte auslesen, mit dem zu PLOTTenden Punkt OR-verknüpfen und dann das Byte wieder 'reinschreiben.

Die Lösung des Problems ist bereits in meinem Programm D/E (s. Info 4) vorhanden, aber dort leider nicht kommentiert. Auch habe ich diesmal die Plotposition angepaßt, will sagen: Das VRAM hat 0,0 oben links, während wir ja 0,0 als unten links interpretieren. Dabei bleibt's jetzt; 0,0 ist unten links. Mein Assemblerprogramm hab' ich kommentiert, aber der Studierende sollte sich des VRAM-Organisations-Blattes bedienen, das K. Muerling mal gemacht hat (Info-Nr. weiß ich nicht).

Das Programm in Zeile 10, das bewußt mit RETURN in Zeile 20 abgeschlossen wurde, damit man es als Subroutine ausprobieren kann, plottet im Hintergrund; man kann also dabei einen Textschirm betrachten. Zeile 30 schaltet aber auf den VS 4 und sorgt auf diesem für Kontrast, damit man gut zugucken kann. Das kann man natürlich weglassen, oder die Farbwerte gesondert via Assembler einschreiben, oder.... Zeile 40 zeichnet einen Balken von links unten nach rechts oben, als Demo. Das ganze einfach mit RUN starten. Die Zeile 0 nicht verändern und auch keine Zeile 1 bis 9 einfügen, sonst muß 10 neu assembliert werden.

```
0 GOTO 30
10 CODE
```

```
800F      JR PLOT
8011 PLOTH: DB 0      ; ;Horizontale PLOT-Position. Erstmal egal.
8012 PLOTV: DB 0      ; ;Vertikale PLOT-Position. Wird auch gePOKET.
8013 PLOT: LD A,(PLOTH) ; ;PLOTH wird später Low-Byte der Adresse.
8016      AND #F8      ; ;Die 3 untersten Bits sind irrelevant für PLOTH.
8018      LD L,A        ; ;HL wird später die Adresse im VRAM.
8019      LD A,(PLOTV)  ; ;Jetzt das Hi-Byte bearbeiten und die
801C      NEG          ; ;PLOT-Position 0,0 nach links oben holen.
801E      ADD A,191
8020      LD H,A        ; ;Und als das Hi-Byte in H ablegen.
8021      AND 7         ; ;Die 3 untersten bits von Hi werden zu den
8023      ADD A,L       ; ;drei untersten von Low. OR ginge auch, weil
8024      LD L,A        ; ;die ja durch AND #F8 weg sind. Neues Low-Byte!
8025      SRL H         ; ;Von Hi-Byte braucht man nur die oberen 5 Bits,
8027      SRL H         ; ;die reichen aus f. 23*256 Bytes. 24 Zeilen hat
8029      SRL H         ; ;der Schirm und Position 0,0 zählt ja mit.
802B      DI
802C      CALL OUTHL    ; ;Adresse ist jetzt in HL zusammengesetzt.
802F      IN A,(1)      ; ;Und von da hol' ich mir ein Bildschirmbyte.
8031      LD C,A        ; ;In C zwischenlagern. Mit C passiert noch 'was.
8032      SET 6,H       ; ;Aber erst den VDP auf 'Schreiben' einstimmen,
8034      CALL OUTHL    ; ;und dasselbe HL als Adresse 'rausschicken.
```

```

8037      LD A,(PLOTH)      ;Jetzt muß ich dieses Byte C verändern. Und das
803A      AND 7            ;richtet sich nach diesen alleruntersten 3 Bits,
803C      XOR 7            ;die ich aber noch umdrehen muß, weil wenn da
803E      RLCA             ;7 steht, ist 0 gemeint. Und Bit 0 ist rechts.
803F      RLCA             ;Daraus konstruiere ich ein SET X,A, das ich
8040      RLCA             ;dann mit C verknüpfe. SET X,A ist binär
8041      OR #C7           ;11001011 und 11XXX111.
8043      LD (SETXA),A
8046      XOR A            ;A löschen und SET X,A ausführen.
8047      DB #CB
8048 SETXA: DB #C7
8049      OR C              ;Die PLOT-Verknüpfung! XOR-PLOT geht so auch.
804A      OUT (1),A        ;Byte wieder auf den Schirm bringen. THAT'S IT.
804C      EI              ;'Cursor' ist lat. und heißt 'Reisender', oder?
804D      RET
804E OUTHL: LD A,L
804F      OUT (2),A
8051      LD A,H
8052      OUT (2),A
8054 WAIT: EX (SP),HL      ;Ein bißchen Zeit verschwenden. Hab' ich jetzt
8055      EX (SP),HL      ;nicht nachgerechnet.
8056      RET

```

Symbols:

```

PLOTH  8011      PLOTV  8012
PLOT   8013      OUTHL  804E
WAIT   8054      SETXA  8048

```

```

20 RETURN
30 VS 4: COLOUR 2,1: COLOUR 3,15: CLS
40 CODE

```

```

8714      LD B,66          ;          ;Das wird ein Balken von li.u.n.r.o.
8716 HELIUM: LD C,192      ;          ;Mal was anderes als immer diese
8718 NEON:  LD A,C         ;          ;LOOPS und LOOPs usw.
8719      ADD A,B
871A      SUB 3           ;          ;Die Berechnung ist zwar etwas
871C      LD L,A          ;          ;verquer, dafür aber kurz.
871D      LD H,C
871E      DEC H
871F      LD (#8011),HL
8722      PUSH BC
8723      CALL #8013
8726      POP BC
8727      DEC C
8728      JR NZ,NEON
872A      DEC B
872B      DJNZ HELIUM     ;STEP-2, damit's streifig wird.
872D      RET

```

Die Berechnung der VRAM-Adresse aus gegebener Plotposition H,V hat ja schon so manchem Rätsel aufgegeben. Hier der in Zeile 10 verwendete (und von #8013 bis #8029 gehende) Algorithmus in BASIC und PASCAL. Die von R. Gmeinwieser in Info 7 vorgeschlagene Lösung ist zwar etwas umständlicher, hat aber die selben Ergebnisse wie die folgenden Lösungen.

```

BASIC:  LET ADR=256*INT((191-V)/8)+H-MOD(H,8)+MOD(191-V,8)
PASCAL: ADR:=256*((191-V) DIV 8)+H-(H MOD 8)+((191-V) MOD 8))

```

TYPEWR

heißt das Programm am Schluß dieses Artikels, und wer jetzt vermutet, es handle sich dabei um eine Schreibmaschine, der irrt nicht. Gedacht ist es für die Leute, die zwar einen Drucker, aber weder Floppy noch Newword haben und ab und zu auch mal in den Genuß so erlesener Dinge wie Randausgleich und Zentrierung kommen wollen. Das ist der Zweck dieses Programms, das für BASIC gar nicht langsam ist. Zur Bedienung. Ab Zeile 110 stehen die sogenannten defaults, Voreinstellungen von Rändern, Modi, usw. Wichtig ist WD; für 40-Zeichen-Schirme (also MTX ohne FDX) muß hier eine 40 stehen, für 80 Zeichen eben 80. LM ist der linke Rand; der steht z.Zt. auf 9, also sind links 8 Leerzeichen. Der rechte Rand (RM) ist so eingestellt, daß die effektive Druckzeile 65 Zeichen breit ist (ohne Ränder, die das Programm ja auch liefert). Mit Rändern ist die Zeile LW Zeichen breit.

Randausgleich und Zentrierung beziehen sich auf die Zeichen zwischen den Rändern, EOL, DEL und INS-Modus beziehen sich jedoch auf den Zeilenrest. Das sollte man beachten, wenn man etwas auf den rechten Rand geschrieben hat. Der Randausgleich gehorcht folgenden Gesetzen. Befindet sich ganz rechts (Position RM) ein Zeichen, dann findet kein Ausgleich statt (z.B. Datumsangabe rechts). Ansonsten werden alle Leerzeichen zwischen LM und dem ersten Zeichen (falls da Leerzeichen sind) entfernt, und der Randausgleich wird vorgenommen. Die Wirkung der Formatierung (Randausgleich oder Zentrierung) kann man sich anschauen, bevor die Zeile gedruckt wird, wenn man will. Dazu auf die Pfeil-hoch-Taste drücken. Es wird gedruckt ohne zu formatieren, indem man auf LINE FEED oder die Pfeil-runter-Taste drückt. RET formatiert und druckt dann gleich. Der Randausgleich dauert um so länger, je mehr Leerzeichen einzufügen sind.

Will man Text einrücken, dann kann man statt der Leerzeichen eine beliebige F-Taste benutzen (BS=1). Dies schafft binding spaces, das Programm sieht dann ein einziges Wort und nicht die Leerzeichen. Die Druckroutine wandelt diese F-Tasten-Zeichen dann wieder in normale Leerzeichen um. Ist BS=0 (s. defaults), dann werden die F-Tasten-Zeichen als eben diese ausgegeben (CHR\$(128), (CHR\$(129)...), und das geht dann natürlich schneller.

Mit PAGE kann man umschalten zwischen Randausgleich, Zentrierung und nichts. Probiert mal ein bißchen rum. Viel Spaß.

Zu EXTENDED BASIC

Am Kopf des Programms stehen einige interessante System-Variablen. Hier eine Liste.

E600 BOTTOM: JP STARTUP	ist nach dem ersten Initialisieren nicht mehr verfügbar. Da steht dann JP INIT
E603 USERJP: JP INIT	Hier springt PROC USER hin.
E606 SYSIZE: DW #E5FF	Hält den durch PROC SIZE festgelegten Wert-1.
E608 BUFSIZ: DW #0008	Maximale Größe des Soundbuffers in Blöcken pro Kanal.
E60A ERRNUM: DB #00	Fehlernummer, von PROC ERROR gespeichert. Runtime-Fehler<=15: Bits 4 und 5 sind gesetzt.
E60B ERRPOS: DW #0000	BASIC-Zeilennr., in der der Fehler auftrat.


```

10 REM *****
15 REM *
20 REM *          TYPEWRITER          *
25 REM * WITH FORMATTING ABILITIES *
30 REM *
35 REM * (C) 1985 ANDREAS VIEBKE *
40 REM *
45 REM *****
50 REM
100 DISC SAVE "TYPEWR"
109 REM -----
110 REM -----DEFAULTS:-----
111 LET WD=80: REM --SCREEN-WIDTH-----
112 LET LW=80: REM --LINE-WIDTH-----
113 LET LM=09: REM --LEFT-MARGIN-----
114 LET RM=73: REM --RIGHT-MARGIN-----
115 LET TC=LM: REM --TAB-COLUMN-----
116 LET IN=01: REM --INSERT-MODE-0=OFF-
117 LET CL=LM: REM --CURSOR-COLUMN-----
118 LET MD=01: REM --MODE 1=ADJ, 2=CTR-
119 LET DL=30: REM --DELAY-----
120 LET BS=01: REM --FKEY=BIND.-SPACE--
121 REM -----
130 DIM L$(LW),I$(1),MD$(10): LET LL=LM
: LET RR=RM
135 FOR I=1 TO LW: LET L$(I)=" ": NEXT
140 LET MD$="          ": IF IN=1 THEN
LET MD$(1)="INS"
145 IF MD=1 THEN LET MD$(6)="ADJ" ELSE
IF MD=2 THEN LET MD$(6)="CTR"
150 LET ST$="COL:      "+MD$+"LM:      RM
:      "
155 LET ST$(5)=STR$(CL): LET ST$(23)=ST
R$(LM): LET ST$(31)=STR$(RM)
160 CRVS 3,-(WD=80)*3,-(WD=40),18,WD-1,
1,WD: PRINT CHR$(29);
165 IF WD=40 THEN PAPER 5: INK 15
170 CRVS 0,-(WD=80)*3,0,20,LW,1,WD: PRI
NT CHR$(29);CHR$(30);
175 IF WD=40 THEN PAPER 5: INK 15
180 GOSUB 6000: VS 3: PRINT ST$;: VS 0
185 REM -----
200 GOSUB 7000
250 VS 3: CSR 4,0: PRINT CL;" ";: VS 0
300 REM -----FETCH-KEY-----
310 PAUSE ABS(DL)+1
320 LET I=ASC(INKEY$): IF I<0 THEN GOTO
320
330 IF I>32 AND I<>127 THEN GOTO 2000
340 PAUSE 10: IF I=32 THEN GOTO 2000
350 IF I=28 OR I=29 THEN GOTO 1000
360 IF I=13 THEN GOTO 4000
370 IF I=8 THEN GOTO 1100
380 IF I=25 THEN GOTO 1200
390 IF I=26 THEN GOTO 1300
400 IF I=12 THEN GOTO 1400
410 IF I=5 THEN GOTO 1500
420 IF I=21 THEN GOTO 1600
430 IF I=127 THEN GOTO 1700
440 IF I=9 THEN GOTO 1950
450 IF I=10 THEN GOTO 4500
460 IF I=27 THEN GOSUB 7000: GOTO 1800
470 IF I=11 THEN GOSUB 5000: GOSUB 7000
600 GOTO 300
1000 REM -----CENTRE/ADJUST-SWITCH-----
1010 IF MD$(6,3)="ADJ" THEN LET MD$(6)=
"CTR ": GOTO 1040
1020 IF MD$(6,3)="CTR" THEN LET MD$(6)=
"      ": GOTO 1040
1030 IF MD$(6,3)="      " THEN LET MD$(6)=
"ADJ  "
1040 VS 3: CSR 9,0: PRINT MD$;: VS 0: G
OSUB 7000: GOTO 300
1100 REM -----BACKSPACE-----
1110 IF CL<=LM THEN LET CL=RM: GOSUB 70
00: GOTO 250
1120 PRINT CHR$(I);: LET CL=CL-1: GOTO
250
1200 REM -----FORWARD-----
1210 IF CL>RM THEN GOSUB 7000: GOTO 300
1220 PRINT CHR$(I);: LET CL=CL+1: GOTO
250
1300 REM -----HOME-----
1310 LET CL=LM: CSR LM-1,0: GOTO 250
1400 REM -----CLEAR-MARGINS-----
1410 LET LM=1: LET RM=LW: GOSUB 7000
1420 VS 3: CSR 22,0: PRINT LM;" ";: CSR
30,0: PRINT RM;" ";: VS 0: GOTO 300
1500 REM -----EOL-----
1510 PRINT CHR$(-5*(CL<=LW));: GOTO 300
1600 REM -----INSERT-KEY-----
1610 LET IN=-(IN=0): IF IN=0 THEN LET M
D$(1)="      " ELSE LET MD$(1)="INS"
1620 VS 3: CSR 9,0: PRINT MD$;: GOSUB 7
000: GOTO 300
1700 REM -----DELETE-KEY-----
1710 PRINT CHR$(-22*(CL<=LW));: GOTO 30
0
1800 REM -----ESC:-MARGINS-AND-TAB-----
1810 LET I$=INKEY$: IF I$="" OR I$=CHR$(
27) THEN GOTO 1810
1820 IF I$="l" OR I$="L" THEN GOTO 1880
1830 IF I$="t" OR I$="T" THEN GOTO 1900
1840 IF I$="m" OR I$="M" THEN GOTO 1920
1850 IF I$<>"r" AND I$<>"R" THEN GOTO 3
00
1860 IF CL>LM THEN LET RM=CL: GOSUB 700
0
1870 VS 3: CSR 30,0: PRINT RM;" ";: VS
0: GOTO 300
1880 IF CL<RM THEN LET LM=CL: GOSUB 700
0
1890 VS 3: CSR 22,0: PRINT LM;: VS 0: G
OTO 300
1900 IF CL>RM THEN GOTO 300
1910 LET TC=CL: GOSUB 7000: GOTO 300
1920 LET LM=LL: LET RM=RR: IF CL<LM OR
CL>RM THEN LET CL=LM
1930 VS 3: CSR 22,0: PRINT LM;" ";: CSR
30,0: PRINT RM;" ";

```

```

1940 VS 0: GOSUB 7000: GOTO 250
1950 REM -----TAB-KEY-----
1960 IF TC<LM OR TC>RM THEN GOSUB 7000:
  GOTO 300
1970 LET CL=TC: CSR CL-1,0: GOTO 250
2000 REM -----CHARACTER-----
2010 IF CL>RM THEN GOSUB 7000: GOTO 300
2020 LET J=21*IN: PRINT CHR$(J)+CHR$(I)
+CHR$(J);: LET CL=CL+1: GOTO 250
4000 REM -----PRINT-----
4010 GOSUB 5000
4500 GOSUB 8000: IF BS=0 THEN LPRINT L$
: GOTO 4550
4510 FOR I=1 TO LW: IF L$(I)>CHR$(127)
THEN LET L$(I)=CHR$(32)
4520 LPRINT L$(I);: NEXT : LPRINT
4550 LET CL=LM: CLS : CSR CL-1,0: GOTO
200
5000 REM -----FORMAT-----
5010 IF MD$(6)=" " THEN RETURN
5020 VS 7: CLS : PRINT "wait.": VS 0
5030 GOSUB 8000: LET LS=LM: LET RS=RM
5040 IF MD$(6)="C" THEN GOTO 5500
5050 GOSUB 5900: IF LS>RM THEN RETURN
5060 IF L$(LM)=" " THEN LET L$(LM)=MID$(
L$,LM+1,RM-LM)+" ": GOTO 5060
5080 IF L$(RS)=" " THEN LET RS=RS-1: GO
TO 5080
5090 LET RC=RM-RS: LET LS=LM: LET RD=RC
: LET RT=RS
5100 GOSUB 5800: IF RS<=LM OR RC=0 THEN
GOTO 5120
5110 LET L$(RS)=" "+MID$(L$,RS,RM-RS):
LET RC=RC-1
5120 GOSUB 5900: IF LS>RM OR RC=0 OR LS
>RM-RC THEN GOTO 5200
5130 LET L$(LS)=" "+MID$(L$,LS,RM-LS):
LET RC=RC-1
5200 IF RC=RD OR RC=0 THEN CSR 0,0: PRI
NT L$;: RETURN
5210 IF LS<RM OR RS>LM THEN GOTO 5100
5220 LET RS=RT: LET LS=LM: GOTO 5100
5500 REM -----CENTRE-----
5520 IF RS<LM THEN RETURN
5530 IF L$(RS)=" " THEN LET RS=RS-1: GO
TO 5520
5540 IF L$(LS)=" " THEN LET LS=LS+1: GO
TO 5540
5550 LET RC=RM-RS: LET RD=LS-LM
5560 IF RD+RC<2 OR ABS(RC-RD)<2 THEN RE
TURN
5570 LET I=INT(ABS(RC-RD)/2): IF RC>RD
THEN GOTO 5600
5580 FOR RC=1 TO I: LET L$(LM)=MID$(L$,
LM+1,RM-LM)+" "
5590 NEXT : GOTO 5700
5600 FOR RC=1 TO I: LET L$(LM)=" "+MID$(
L$,LM,RM-LM): NEXT
5700 CSR 0,0: PRINT L$;: RETURN
5800 REM -----<-FIND-NEXT-WORD-----
5810 IF RS<LM THEN RETURN
5820 IF L$(RS)=" " THEN LET RS=RS-1: GO
TO 5810
5850 REM -----<-FIND-NEXT-SPACE-----
5860 IF RS<LM THEN RETURN
5870 IF L$(RS)=" " THEN RETURN
5880 LET RS=RS-1: GOTO 5860
5900 REM -----FIND-NEXT-WORD-->-----
5910 IF LS>RM THEN RETURN
5920 IF L$(LS)=" " THEN LET LS=LS+1: GO
TO 5910
5950 REM -----FIND-NEXT-SPACE->-----
5960 IF LS>RM THEN RETURN
5970 IF L$(LS)=" " THEN RETURN
5980 LET LS=LS+1: GOTO 5960
6000 REM -----MENU-----
6010 VS 5: CLS
6020 PRINT "      TYPEWRITER (C) EDIT KEYS": PR
INT
6030 PRINT "->                move cursor right"
6040 PRINT "<-                move cursor left"
6050 PRINT "DEL            delete char at cursor pos"
6060 PRINT "EOL            erase to end of line"
6070 PRINT "HOME          cursor to left margin"
6080 PRINT "ESC l/r        set left/right margin"
6090 PRINT "ESC t          set tab stop"
6100 PRINT "ESC m          reset default margins"
6110 PRINT "TAB            move cursor to tab stop"
6120 PRINT "CLS            clear both margins"
6130 PRINT "INS            toggle insert mode"
6140 PRINT "PAGE          switch ADJ,CTR,no ADJ"
6150 PRINT "LFEEED        print without formatting"
6160 PRINT "RET            format and print"
6170 PRINT "~            format without printing"
6500 VS 7: CLS
6510 PRINT "Copyright (C) 1985 Andreas Viebke"
;
6520 PAUSE 3000: VS 0: RETURN
7000 REM -----BIEP-----
7010 VS 7: CLS : VS 0: CSR CL-1,0
7020 SOUND 0,75,15: PAUSE 100: SOUND 0,
0,0: RETURN
8000 REM -----READ-SCREEN-----
8010 CSR 0,0: FOR I=0 TO LW-1: LET L$(I
+1)=SPK$(
8020 NEXT : CSR 0,0: RETURN
9000 REM -----END-----

```

das ist natürlich
mehr als flüssig.
-Anm. d. Autors

1. Zu RAM-Disc

Nach Bestückung der 32K Speichererweiterung mit 8 billigst 4164 und Abschnipeln von 3 Lötkelchen der RS232-0 Buchse habe ich also nun eine 80K RAM-Disc(abz. System 7K, Directory 1K).

Darüber laufen bei mir im Moment SUPERCALC und TURBO-Pascal, nach dem Erweitern der Speicherkarte auf 128K ist dann auch NEWWORD auf der RAM-Disc lauffähig(benötigt 134K).

Dabei ist es sinnvoll, auf der RAM-Disc nur die Programme und Overlays zu halten, alle zu verändernden eigenen Daten gehören auf eine echte Diskette; denn vergessen ist menschlich...

Über den Geschwindigkeitsvorteil der RAM-Disc braucht man wohl nichts zu sagen, er ist erheblich. So sind bei SUPERCALC und TURBO die Zugriffe auf die Overlays nicht mehr spürbar und es ist beeindruckend, wie z.B. auf der RAM-Disc vorhandene *.XQT-Files von SUPERCALC abgearbeitet werden.

2. Terminal-Installation TURBO-Pascal (bei mir 3.0)

Da kam es nach langem Warten nun bei mir an: TURBO-Pascal 3.0 im Televideo DS/DD Format. Also schnell auf Laufwerk C diese Format eingestellt und auf MTX-Format kopiert, Dank Info's kein Problem. Danach ein kurzer Blick ins Handbuch, mutig TINST aufgerufen, Handbuch TURBO links, Manual 80 Zeichen-Karte rechts, Tastatur in der Mitte, das kann denn doch wohl kein so großes Problem sein!

Aber wie immer, der Teufel steckt im Detail, zugeschlagen hat er bei mir bei CONTRL-K (Pfeil nach oben), damit waren dann alle Block-Kommandos nicht verfügbar, die ja auch mit CONTRL-K beginnen. Des weiteren Probleme mit HOME=CONTRL-Z, sowie Cursor nach Rechts=CONTRL-Y; da möchte TURBO gern eine Zeile löschen.

Nach einigem Hin und Her erfolgte dann das Abschalten des MTX und die Problemlösung mit Bleistift und Papier. Hier nun alle Angaben zu TINST so, wie ich sie durchgeführt habe:

- Screen Installation (Kennung in Bezug auf Handbuch):

Wahl: None of the above, da Heimsoeth das Memotech-Terminal nicht kennt

- L.2.1. Memotech FDX System<RET>
- L.2.2. N<RET>
- L.2.3. Y<RET>
- L.2.3.1. <CTRL>L<RET> = Clear Screen
- L.2.4. <CTRL>C<RET>
- L.2.4.1. <RET>
- L.2.4.2. <RET>
- L.2.4.3. Y<RET>
- L.2.4.4. 32<RET>
- L.2.4.5. 32<RET>
- L.2.4.6. Y<RET>
- L.2.5. <CTRL>L
- L.2.6. Y<RET>
- L.2.7. <ESC>J<RET>
- L.2.8. <ESC>I<RET>
- L.2.9. <CTRL>E<RET>
- L.2.10.1. <CTRL>F<CTRL>D<RET>
- L.2.10.2. <CTRL>F<CTRL>S<RET>
- L.2.11. 24<RET>
- L.2.12. 80<RET>
- L.2.13.1. <RET>
- L.2.13.2. <RET>
- L.2.13.3. <RET>
- L.2.14. Y<RET>
- L.2.15. 4<RET>

- Command Installation:

1	<RET>	23	<CTRL>N
2	<RET>	24	<EOL>
3	<CTRL>Y	25	<RET>
4	<RET>	26	
5	<RET>	27	<RET>
6	<CTRL>K	28	<RET>
7	<LINE FEED>	29	<CTRL>B<CTRL>B
8	<RET>	30	<CTRL>B<CTRL>K
9	<RET>	31	<CTRL>B<CTRL>T
10	<RET>	32	<CTRL>B<CTRL>H
11	<RET>	33	<CTRL>B<CTRL>C
12	<RET>	34	<CTRL>B<CTRL>V
13	<RET>	35	<CTRL>B<CTRL>Y
14	<HOME>	36	<CTRL>B<CTRL>R
15	<RET>	37	<CTRL>B<CTRL>W
16	<RET>	38	<CTRL>B<CTRL>D
17	<RET>	39	<TAB>
18	<RET>	40	<RET>
19	<RET>	41	<RET>
20	<RET>	42	<RET>
21	<INS>	43	<RET>
22	<RET>	44	<RET>
		45	^

Beenden mit Q.

3. Belegen der Funktionstasten:

Da der Turbo-Pascal-Editor weitgehend WORDSTAR- (und somit auch NEWWORD-) kompatibel ist, habe ich die Funktionstastenbelegung so weit wie möglich identisch wie bei NEWWORD angelegt. Das entsprechende TURBO.KBD -File wird mittels F.COM bzw. RAM F initialisiert.

4. Generelles zu TURBO:

Ich bin damit sehr zufrieden, insbesondere wenn ich an FDX-BASIC denke. Allerdings zwingt einem PASCAL eben dazu, sich vorab Gedanken über ein Programm zu machen, es zu strukturieren, während man in BASIC halt einfach munter drauf los programmiert und das Ding schon irgendwie zum laufen bringt. Natürlich geht nicht jeder so vor, aber die Gefahr ist groß.

Soviel für heute.

Lesen von Alpatronik-Disketten (P 2 und P3)

Da ich nur ein 40-Spur-Laufwerk besitze, gehe ich folgendermaßen vor:

Beim Alpatronik P 3: Kopieren auf Laufwerk P:

Diese Diskette im FDX lesen nach Aufruf von P2FMT (OVERLAY8.COM mit folgendem DPH (streng nach Diskettenstory!):

0F19:20/00/03/07/00/97/00/3F/00/C0/00/20/00/02/00)

und CFIGP2 (CFIGB mit geänderter Bezeichnung: TELEVIDE0-->P2FMT)

B:03 C:08, FDX sieht hier eine einseitige 40-Spur-Diskette mit 150 K. Mit COPY B:03 C:08 kann alles kopiert werden. Auf C: kann auch gearbeitet werden.

Die unterschiedliche Extent-Bezeichnung ist nach meinen Erfahrungen unerheblich. (1. Ext. -->00 oder 01).

Alle CP/M-Programme laufen. Programme mit Bildschirm-Steuerung müssen angepaßt werden!

Die anliegende Übersetzungsliste enthält mit Sicherheit noch Fehler! Leute mit Erfahrung (ich gehöre nicht dazu!) werden gebeten, diese Liste zu überarbeiten.

Z. Zt. suche ich mühsam mit DDT die infragekommenden 1BH ff und tausche sie gegen unsere Steuerzeichen. Rezepte willkommen !!!

Kann evtl. NWINSTAL.COM umfunktioniert werden?

Umbau MTX nur wegen heißem Innenleben und warmem Gehäuse?

Bei meinem MTX hatte die Kühlfläche der Spannungsregler nur über die Halteschraube Kontakt mit dem Gehäuse. Hohe Temperaturen waren die Folge.

Einfachste Abhilfe:

Ausbau Hauptplatine, Abkanten der Kühlfläche nach unten, so daß sie nach Wiedereinbau satt auf der Unterschale des Gehäuses aufliegt. Dies geht einfach, da weiches Alu verwendet wurde; ein Beschädigen der Platine ist unwahrscheinlich.

Beim anschließendem Einschieben der Platine ins Gehäuse reichlich Wärmeleitpaste verwenden.

Erfolg: Die Unterschale wird, nach gehörigem Festziehen der Befestigungsschraube, nur noch handwarm, die Oberschale behält die Zimmertemperatur.

Tip: Unterlegen von irgendwas hinten unter den MTX erhöht den Schreibkomfort und läßt die reichliche kühle Abluft des FDX teilweise unter den MTX streichen.

Dennoch sollten manche PAL's und PROM's Kühlkörper erhalten.

H. Traber

Programmieren in Forth (oder wie man Hürden nimmt)

Nachdem ich das Forth, von dem ich schon soviel Positives hörte, endlich im Laufwerk hatte, dieses mit Forth (oder Z80FORTH) startete, meldete es sich auch tatsächlich. Es meinte 'No File', 'Z80-fig-forth 1.1g'. Anschließend nahm ich mir mein Buch und wollte wild loslegen, es klappte auch so ziemlich alles, nur konnte ich keine Diskettenoperationen ausführen. Ich konnte zwar eine SCR-nr. eingeben und dann LOAD RET. drücken, aber es wurde halt nichts geladen. Nur weil ich beim Starten von Forth nicht direkt den Filenamen meines SCR-files mitangegeben habe (aber bis ich das mal raus hatte).

Also starten des Forth mit A>FORTH A:file.ext (z.B. A: SCREENS.FRT). Ich habe es bisher geschafft nur mit Screens.frt auszukommen, wenn man mehr Platz braucht, kann das File mit EXTEND vergrößert werden (z.B.: 10 EXTEND erweitert das angegebene Screenfile um 10 Blocks), wenn auf der Diskette genug Platz ist. Nach diesen anfänglichen Schwierigkeiten war es mir dann endlich möglich den Editor, Asssembler und den Deforth-er zu nutzen. Wie der EDITOR gestartet wird, wurde ja schon im letzten Info erwähnt. Wie der EDITOR, der übrigens (zumindest meiner) manchmal seine Macken hat, bedient wird, steht in EDITOR.DOC. Sollte dieses File nicht vorhanden sein, hier nun kurz die wichtigsten Befehle.

Durch Zeilennr. P Text wird der hinter P stehende Text in der Zeile Zeilennr. eingeben. Mit L kann der SCR gelistet werden. Mit SCR-nr. CLEAR wird ein SCR gelöscht, und mit FLUSH gespeichert. Dies in aller Kürze zum EDITOR. Ab SCR 13 ist ja ein ASSEMBLER in 8080-Code mit Z80-Erweiterungen. Neben den üblichen Mnemonics bietet er auch die Möglichkeit bedingte Schleifen einzubinden. Im Unterschied zu Colon-wörtern, also Wörter, die mit einem ':' eröffnet werden, beginnt ein Assembler-Wort mit CODE WORDMnemonics.....C; . Man gibt also mit dem Editor eine Code-definition (in einen SCR) ein, lädt dann den ASSEMBLER (mit 13 LOAD), startet den Assembler mit ASSEMBLER RETURN, lädt den SCR mit der Assembler-routine und wenn man dann nichts falsch gemacht hat Aber mit dem Assembler hab' ich noch keine großen Erfahrungen gemacht.

Ganz nützlich ist dagegen der ' DEFORTHER ', der beginnt bei SCR 34 (mit 1 LIST bekommt man eine Art Inhaltsverzeichnis von SCREENS.FRT). Ist der Deforth-er geladen, kann man sich jedes Colonwort wieder anschauen. Man muß dazu nur DEFORTH WORT eingeben. Dies ist besonders dann von Nutzen, wenn man Wörter mit dem Zeileneditor eingegeben hat, da diese dann kompiliert werden und der Sourcecode verloren geht. Auch das Eingeben von Wörtern (mit dem Zeileneditor), die länger als eine Zeile sind, ist möglich. Man muß dazu nur nach dem Return die neue Zeile mit ü (oder eckige Klammerzu, je nach Zeichensatz) eröffnen.

Die von Frank Dersewski angebotene Forthversion kommt ja ursprünglich von Edmund Ramm, ich weiß nicht ob Frank schon die neuere Version hat, (meldet sich mit Figforth V1.1h). Im Unterschied zur g-Version sind die folgenden Befehle neu dazu gekommen: 0<>, 0> TUCK, -ROT, NIP, CSWAP PICK, -CMOVE. Bis jetzt hab ich nur die Funktion von PICK (0<>, 0> sind ja klar) heraus. PICK kopiert den n-ten Wert auf den Stack, also 1 PICK entspricht DUP, 2 PICK -> OVER. Wenn jemand Interesse an dieser Version des Forth hat soll er mir bitte schreiben (Diskette + Rückporto).

An Literatur zu Forth kenne ich einmal von R. Zech ' Die Programmiersprache Forth ' oder von L. Brodie ' Programmieren in Forth '. Für die,

die lieber eng beim Figforth bleiben wollen, und einen teilweise trockenen Stil nicht scheuen, ist wohl Zech der richtige Autor. Brodie dagegen stellt Forth recht anschaulich dar, durch Cartoons wird der Stoff witzig dargestellt und man kann das Buch gut lesen. Viel Literatur und Software bekommt man bei:

Forth-Systeme Angelika Flesch
Postfach 1311
7820 Titisee-Neustadt.

Mit Unterstützung des Installationmanuels und eines Artikels aus der c't (10,84) versuche ich gerade eine Ramdisk zu simulieren, die aber innerhalb der 64k liegt, also kein Ersatz für das Programm von Bernd Preusing ist.

Ein NOT kann man sich folgendermaßen einbauen:

```
: NOT 0= IF 1 ELSE 0 THEN ;.
```

19 LIST
SCR# 19

```
0 ( ***** DUMP nach R. Zech ***** )
1 HEX ( 1000 1010 DUMP gibt Speicherinhalt von 1000-1010 aus )
2 : DUMP
3   CR CR BASE $ HEX      ( vereinbare HEX, rette Basis      )
4   ROT ROT               ( LOOP-Parameter wieder oben      )
5   1+ SWAP              ( Start/Endadresse inclusive      )
6   DO
7     I 0 8 D.R          ( drucke Adresse rechtsbündig      )
8     3 SPACES 3A EMIT  ( Spaltenbegrenzung, ASCII ,:'    )
9     SPACE SPACE       ( 2 Leerspalten                    )
10    I 4 0 DO
11    DUP I + C$        ( lies Bytes $ Adresse für        )
12    4 .R              ( eine Zeile und drucke rechts-    )
13    LOOP              ( bündig aus                       )
14    DROP CR 4 +LOOP   ( vermeide Bild 12; CR -> neue Zeile )
15    BASE ! ;
```

ok

21 LIST
SCR # 21

```
0 ( ***** ZUFALLSZAHLENGENERATOR ***** )
1
2 0 VARIABLE RND  HERE RND !
3
4 : RANDOM RND $ 31421 * 6927 + DUP RND ! ;
5 : ZUFALLSZAHL RANDOM U* SWAP DROP ;
6
7 ( ZUFALLSZAHL ERWARTET EINE ZAHL AUF DEM STACK UND ERZEUGT )
8 ( DANN EINE ZUFALLSZAHL BSP.: 10 ZUFALLSZAHL )
9 ( NACH LEO BRODIE, PROGRAMMIEREN IN FORTH )
10
11 : NOT 0= IF 1 ELSE 0 THEN ;
12
13
14
15
```

ok

(Anm.d. Red.: Ein bißchen Textformatierung würde auch nicht schaden)

Ansteuerung des MTX-Tongenerators unter Pascal

Nachdem bereits Olaf Krumnow in Info 6 gezeigt hat, wie man Pascal zur Erzeugung von VS4-Grafiken verwenden kann, fehlt eigentlich nur noch die Soundgenerator-Ansteuerung, um das FDX-Basic endgültig vergessen zu können. Hier werden nun Basisroutinen zur Tonerzeugung unter Pascal vorgestellt.

1. Mechanismus der Datenübergabe an den Tongenerator

Wie im MTX-Handbuch beschrieben (S.228-232), muß zunächst über Port 6 ein Datenwort an den Tongenerator ausgegeben werden, das von diesem jedoch erst in seine internen Register übernommen wird, nachdem er ein 'strobe'-Signal erhalten hat. Dieses Signal wird von der CPU durch einen Lesebefehl von Port 3 erzeugt. Dabei wird nur das Port-Selektionssignal zur Aktivierung des Soundgenerator benutzt, die (wie bei jedem Port-Lesebefehl) vom Datenbus 'eingesammelten' Signale interessieren hier nicht.

Damit dieses Verfahren reibungslos funktioniert ist folgendes zu beachten :

- nach jeder Datenausgabe über Port 6 muß ein Strobe-Signal erzeugt werden (sonst droht Verlust der übertragenen Daten bei Überschreiben durch ein neues Datenwort)
- zwischen je zwei Strobe-Signalen müssen mind. 32 Taktperioden liegen. (diese Zeit benötigt der Tongenerator zur Übernahme der Daten in die internen Register)

2. Formate der Daten für den Tongenerator

2a) Frequenzfestlegung für die Tonkanäle

Die Frequenz der zu erzeugenden Schwingung wird dem Tongenerator in einem 10-Bit Datenwort mitgeteilt. Schon hieraus ersieht man, daß man mit der Übertragung eines Datenbytes nicht auskommt. Außerdem muß dem Generator noch mitgeteilt werden, für welchen Kanal die Angabe gemeint ist.

Datenformat Frequenzfestlegung :

D7:D6:D5:D4:D3:D2:D1:D0

---+---+---+---+---+---+---+---+---+---

1. Byte 1:R2:R1:R0:F3:F2:F1:F0

2. Byte 0: X:F9:F8:F7:F6:F5:F4

dabei bedeuten:

R_i : Register- (= Kanal-) Auswahl (s.MTX-Handbuch S.231)

F_i : Frequenzwert (s. MTX-Handbuch S.182)

X : 'don't care '

Wie man sieht, werden die niederwertigsten 4 Bits des Frequenzwertes im 1. Byte übertragen, der Rest im 2. Byte. Daher muß eine Aufteilung vorgenommen werden, die in Pascal wie folgt aussieht :

Sei $f = (f_9, \dots, f_0)$ eine ganze Zahl, die sich durch 10 Bits darstellen läßt (also aus dem Bereich 0..1023), dann ist

$f_{low} = (f_3, \dots, f_0) = f \bmod 16$ und

$f_{high} = (f_9, \dots, f_4) = f \text{ div } 16$.

Weiter muß die Kanal-Angabe die richtige Wertigkeit im Datenwort erhalten:

Sei `type tonkanal = (I,II,III);`

`var k : tonkanal;`

dann ist

$r = (r_2, r_1, r_0) = 16 * 2 * \text{ord}(k)$

Insgesamt ergibt sich für die Steuerworte

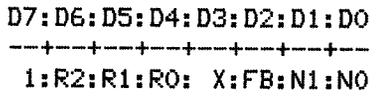
1. Byte $128 + 16 * 2 * \text{ord}(k) + f \bmod 16$

2. Byte $f \text{ div } 16$

2b) Kontrolle des Rauschkanals

Hier gibt es zwei einstellbare Parameter: Shift - Rate und Rauschart (s. MTX-Handbuch S.229/230)

Diese Angaben haben, zusammen mit der Kanalauswahl, in einem Byte Platz:



- mit :
- R_i : Registerauswahl
- FB : Rauschart
- N_i : Shiftrate
- X : 'don't care'

```

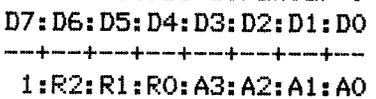
Mit type shiftrate = 0..3;
      rauschart = (p,w);
var   a : rauschart;
      sr : shiftrate;

```

und (R₂,R₁,R₀)= 6 zur Registerauswahl für die Rauschkontrolle ergibt sich in Pascal für das Datenwort 128 + 96 + ord(a)*4 + sr

2c) Lautstärkeeinstellung

Das hierfür benötigte Datenwort hat für alle Kanäle (egal ob Ton- oder Rauschkanal) dieselbe Struktur :



- mit:
- R_i : Registerauswahl
- A_i : Abschwächung des Signals des mit R_i selektierten Kanals in 16 Stufen (A₃,...,A₀) = 0 bedeutet minimale Abschwächung, also maximale Lautstärke, (A₃,...,A₀) = 15 hingegen minimale Lautstärke

Die Pascal-Realisierung dieses Datenworts wird durch

```

128 + 16*(2*ord(k)+1) + 15-l
erreicht, wobei type lautst = 0..15;
var   l : lautst;

```

ist. Hierbei bedeuten große l-Werte auch große Lautstärken und umgekehrt, was 'benutzerfreundlicher' als das Arbeiten mit Abschwächungswerten ist.

3. Realisierung in Pascal

Glücklicherweise unterstützt Turbo-Pascal die Ein-/Ausgabe über Ports direkt durch das vordefinierte Port-Array. Ausgabe über Port 6 wird als

```
port(.6.):=... ,
```

Eingabe über Port 3 als

```
...:= port(.3.)
```

codiert.

Es folgt ein kleines Pascal-Programm, das Routinen zur elementaren Ansteuerung des Tongenerators (vergleichbar dem direkten SOUND-Befehl in BASIC) und einige Anwendungsbeispiele dieser Routinen enthält.

Wolfgang Tesch

Manche Toolbox-Besitzer werden sich schon über die blöde Farbeinstellung des GINST.COM geärgert haben. Vorallem die Besitzer eines Farbmonitors.

GINST.COM erzeugt nämlich zumindest bei meiner Version die gleiche Vordergrund und Hintergrundfarbe.

Fazit man sieht absolut nichts.

Man kann GINST.COM mit sich selber installieren, ist aber schlecht, wenn man die Fragen die gestellt werden nicht sieht.

Am Besten man holt das GINST.COM in den DDT und ändert folgendende Adressen:

```

01C2  02  ändern auf  01
01C3  1B  ändern auf  12
01C8  02  ändern auf  01
01C9  1B  ändern auf  14

```

Bei monochromen Monitoren ist es besser 12 und 14 zu vertauschen.

Das gleiche muß man mit GINST.COD machen, da sonst nach dem ersten Installieren das gleiche Spiel bei dem neuen File los geht.

Nach den Änderungen den DDT jeweils mit ^C verlassen, und mit SAVE #3 GINST.COM und SAVE 91 GINST.COD

Danach ruft man GINST.COM auf.

Es erscheinen folgende Fragen:

Enter name of program to install:

Eingabe:GINST >RET<

Enter firstname for installation files:

Eingabe:TEST >RET<

Danach werden die Files TEST.COM, TEST.DTA UND TEST.M56 erzeugt.

Jetzt ruft man TEST.COM auf, und installiert GINST.COM genauso wie Turbo-Pascal.

Dann wird GINST.DTA gelöscht und TEST.DTA nach GINST.DTA umbenannt.

Ab dann sind die Daten des Memotech im GINST.DTA enthalten.

Eingaben nach Aufruf eines mit GINST.COM erzeugten Installationsfiles, oder beim Installieren mit TINST.COM:

Die Eingaben erfolgen der Reihe nach.

S

31 >RET<

Memotech MTX >RET<

N 2 mal hintereinander

^C >RET<

>RET< 2 mal hintereinander

y

32 >RET< 2 mal hintereinander

y

^L >RET<

y

>ESC< J >RET<

>ESC< I >RET<

^E >RET<

^R >RET<

(<^R und ^T können vertauscht werden je nachdem ob man High- oder Lowvideo haben möchte)

^T >RET<

24 >RET<

80 >RET<

0 >RET< 3 mal hintereinander

y

4 >RET<

Q

Tastaturpatch TAST.COM (Hans-Dieter Müller)

Das folgende Programm führt mehrere Änderungen des Tastaturreivers aus:

- 1. Tastaturentprellung
- 2. Ä, Ö, Ü mit AlphaLock
- 3. Pfeiltasten richtig belegen ^X, ^S, ^D, ^E
- 4. INS-Taste richtig belegen ^V
- 5. Autorepeat schneller
- 6. kleiner Cursor

(Der kleine Cursor ist von Hagen Wenzek, diverse Tastaturadressen von Bernd Preusing)

Das Programm kann im DDT oder VDEB ab 100H eingegeben werden. Im DDT mit 5100 >RET<, Eingabe >RET<... im VDEB mit D100 >RET<, Eingabe >RET<... danach ^C und SAVE 2 TAST.COM

Hexdump zur Eingabe in DDT oder VDEB:

```

-d100
0100 C3 03 02 00 00 00 00 00 00 00 00 00 00 00 00 .....
0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0130 00 00 CD 29 0B FE 29 C4 9D 04 C3 40 0B 3E 01 32 ...).).).s.>.2
0140 2F 3D C3 59 2B AF 32 2F 3D C3 59 2B AF 32 2D 3D /=.Y+.2/= .Y+.2-=-
0150 C3 59 2B CD 55 0B C2 47 2B 0C 0D CA 47 2B F5 CD .Y+.U..G+...G+..
0160 0E 0C C4 24 0C C2 3A 2B 7E F6 40 77 F1 FE 2C C0 ...$....+β.5w...
0170 CD 55 0B C2 C1 04 C3 2C 2B 3E 01 32 2D 3D C9 3E .U.....,+>.2-=-.>
0180 FF 32 2E 3D C3 59 2B AF 32 2E 3D C3 40 0B 3E 01 .2.=.Y+.2.=.s.>.
0190 32 2E 3D C3 59 2B AF C3 7D 2B 3E FF C3 7D 2B 3A 2.=.Y+..ü+>..ü+:
01A0 8D 40 2F 6F 3A EC 3C B7 CA 59 2B 7D 32 8D 40 32 .s/o:.<..Y+ü2.52
01B0 2A 3D C3 59 2B 3E 01 32 EA 3C C3 59 2B AF 32 EA *=.Y+>.2.<.Y+.2.
01C0 3C C3 59 2B 3E FF 32 F2 3C CD C4 28 AF 32 F2 3C <.Y+>.2.<..(.2.<
01D0 7A B7 C2 9D 04 3A 11 3E FE 20 C0 7B 3D F8 CA 9D z.....>..ä=...
01E0 04 FE 10 D2 9D 04 3C 32 F1 3C AF 67 6B C3 34 1B .....<2.<.gk.4.
01E2 CD C4 28 7A B7 C2 9D 04 3A EC 3C B7 C8 7B B7 CA ..(z.....<..ä..
0200 E1 2B FE 11 11 02 CD 19 02 CD 1F 02 0E 00 CD 05 .+.....
0210 00 09 0B 6F 6B 2E 0D 0A 24 0E 09 CD 05 00 C9 3E ...ok...$.>
0220 40 32 1F F0 3E C6 32 8C F1 3E 00 32 65 F2 32 66 52..>.2..>.2e.2f
0230 F2 32 67 F2 3E 18 32 D2 F1 3E 13 32 D5 F1 3E 04 .2g.>.2..>.2..>.
0240 32 D4 F1 3E 05 32 D6 F1 3E 16 32 C9 F1 3E 0A D3 2..>.2..>.2..>..
0250 38 3E 69 D3 39 3E 0B D3 38 3E 09 D3 39 C9 CA 9D 8>i.9>..8>..9...
0260 04 FE 10 D2 9D 04 3C 32 F1 3C AF 67 6B C3 34 1B .....<2.<.gk.4.
0270 CD C4 28 7A B7 C2 9D 04 3A EC 3C B7 C8 7B B7 CA ..(z.....<..ä..
0280 3D =

```

ORG 100H

START:

```

CALL TAST ;Tastaturpatch aufrufen
LD DE,STR ;fertigmeldung laden
CALL STRING ;und aufrufen
LD C,0 ;Warmstartfunktion laden
CALL 5 ;und aufrufen

```

STR:

```

DB 09H,0BA,'ok.',0DA,0AA,'$'

```

STRING:

```

LD C,9
CALL 5
RET

```

TAST:

```

LD A,40H ;Tastaturentprellung
LD (0F01FA),A
LD A,0C6H ;Autorepeat schneller
LD (0F18CA),A
LD A,00H ;AlphaLock mit A,ö,ü
LD (0F265A),A
LD (0F266A),A
LD (0F267A),A
LD A,18H ;Pfeil unten ^X
LD (0F1D2A),A
LD A,13H ;Pfeil links ^S
LD (0F1D5A),A
LD A,04H ;Pfeil rechts ^D
LD (0F1D4A),A
LD A,05H ;Pfeil oben ^E
LD (0F1D6A),A
LD A,16H ;INS ^U
LD (0F1C9A),A
LD A,0AA ;Routine für kleinen Cursor
OUT (38A),A
LD A,69H
OUT (39A),A
LD A,0BA
OUT (38A),A
LD A,09H
OUT (39A),A
RET

```

END START

```

*****

```

PS:

Die Cursor-Routine stammt von Hagen Wenzek.

Diverse Tastaturadressen stammen von Bernd Preusing.

```
1 program set_printer;
2   { Interaktives Setzen einiger Drucker-Parameter
3     (c)'85   Wolfgang Tesch   }
4
5   const nbgrd = ^F#04;
6         bgrd = ^F#16;
7         nbik = ^F#04;
8         blk = ^F#68;
9         con = #30;
10        cso = #31;
11        ESC = #27;
12        CR = #13;
13        FWD = #25;
14        BS = #08;
15
16 type line = string[80];
17       cmd = string[4];
18 var   akt : cmd;
19       str_s, str_1, str_2 : line;
20       escape, bereit, again: boolean;
21       zeich: char;
22
23 {=====}
24
25 procedure realisiere(std, cmd_1, cmd_2: cmd; str_s, str_1, str_2: line);
26   var next: boolean;
27       ch: char;
28
29   begin
30     akt:=std;
31     if escape then write(lst, akt, CR)
32     else begin
33       write(str_s);
34       next:=false;
35       while not next do
36         begin
37           read(kbd, ch);
38           case ch of
39             CR : begin
40                   write(lst, akt, CR);
41                   next:=true;
42                 end;
43             ESC : begin
44                   write(lst, akt, CR);
45                   next:=true;
46                   escape:=true;
47                 end;
48             FWD : begin
49                   write(CR, str_2);
50                   akt:=cmd_2;
51                 end;
52             BS : begin
53                   write(CR, str_1);
54                   akt:=cmd_1;
55                 end;
56           end;
57         end;
58       end;
59     end;
60 {=====}
61 procedure schriftart ;
62   const wert_1 = 'Pica';
63         wert_2 = 'Elite';
64         wert_3 = 'Nerd';
```

```

65      cmd_2    = #27'P'#00;
66      std      = cmd_2;
67      begin
68          str_1:= 'Schriftart : '+bgrd+wert_1+nbgrd+' '+wert_2;
69          str_2:= 'Schriftart : '+wert_1+' '+bgrd+wert_2+nbgrd;
70          str_s:=str_2;
71          realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
72          writeln;writeln;
73      end;
74      {=====}
75      procedure zeichensatz;
76          const wert_1 = 'deutsch';
77                wert_2 = 'amerikanisch';
78                cmd_1  = #27'R'#02;
79                cmd_2  = #27'R'#00;
80                std    = cmd_2;
81      begin
82          str_1:= 'Zeichensatz : '+bgrd+wert_1+nbgrd+' '+wert_2;
83          str_2:= 'Zeichensatz : '+wert_1+' '+bgrd+wert_2+nbgrd;
84          str_s:=str_2;
85          realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
86          writeln;writeln;
87      end;
88      {=====}
89      procedure zeilenzahl;
90          const wert_1 = '72 (12)';
91                wert_2 = '66 (11)';
92                cmd_1  = #27'C'#72;
93                cmd_2  = #27'C'#66;
94                std    = #27'C'#72;
95      begin
96          str_1:= 'Anzahl der Zeilen / Seite : '+bgrd+wert_1+nbgrd+' '+wert_2;
97          str_2:= 'Anzahl der Zeilen / Seite : '+wert_1+' '+bgrd+wert_2+nbgrd;
98          str_s:=str_1;
99          realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
100         writeln;writeln;
101     end;
102     {=====}
103     procedure abstand;
104         const wert_1 = 'normal';
105               wert_2 = 'klein';
106               cmd_1  = #27'2';
107               cmd_2  = #27'A'#08;
108               std    = cmd_1;
109     begin
110         str_1:= 'Zeilenabstand : '+bgrd+wert_1+nbgrd+' '+wert_2;
111         str_2:= 'Zeilenabstand : '+wert_1+' '+bgrd+wert_2+nbgrd;
112         str_s:=str_1;
113         realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
114         writeln;writeln;
115     end;
116     {=====}
117     procedure kursiv;
118         const wert_1 = 'an';
119               wert_2 = 'aus';
120               cmd_1  = #27'4';
121               cmd_2  = #27'5';
122               std    = cmd_2;
123     begin
124         str_1:= 'Kursivschrift : '+bgrd+wert_1+nbgrd+' '+wert_2;
125         str_2:= 'Kursivschrift : '+wert_1+' '+bgrd+wert_2+nbgrd;
126         str_s:=str_2;
127         realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
128

```

```

129  end;
130  {=====}
131  procedure doppel;
132  const wert_1 = 'an';
133        wert_2 = 'aus';
134        cmd_1  = #27'G';
135        cmd_2  = #27'H';
136        std    = cmd_2;
137  begin
138    str_1:='Doppeldruck : '+bgrd+wert_1+nbgrd+' '+wert_2;
139    str_2:='Doppeldruck : '+wert_1+' '+bgrd+wert_2+nbgrd;
140    str_s:=str_2;
141    realisiere(std,cmd_1,cmd_2,str_s,str_1,str_2);
142    writeln;writeln;
143  end;
144  {=====}
145  procedure titel;
146  var i:byte;
147  begin
148    clrscr;
149    write(chr(133));
150    for i := 2 to 79 do write(chr(128));
151    write(chr(132));
152    for i := 2 to 5 do
153      begin
154        gotoxy(1,i);write(chr(129));
155        gotoxy(80,i);write(chr(129));
156      end;
157    write(chr(131));
158    for i := 2 to 79 do write(chr(128));
159    write(chr(130));
160    gotoxy(35,3);write('S E T L S T');
161    gotoxy(18,4);write('Programm zum Veraendern von Druckerparametern');
162    gotoxy(1,7);
163  end;
164  {=====}
165  procedure teste_drucker;
166  var drstatus:byte;
167  begin
168    drstatus:=port[4] and 15;
169    bereit:=drstatus = 10;
170    if not bereit then
171      begin
172        titel;
173        write(' ');
174        writeln(blk,'!!!! Drucker nicht bereit zum Empfang von Daten !!!!',nbk);
175        gotoxy(10,15);
176        writeln('...wartend auf Druckerbereitschaft');
177        repeat
178          drstatus:=port[4] and 15;
179          bereit:=drstatus = 10;
180        until bereit or keypressed;
181        if keypressed then
182          begin
183            clrscr; write(con);
184          end;
185        end;
186  end;
187  {=Hauptprogramm=====}
188  begin
189    write(cso);
190    teste_drucker;
191    if not bereit then exit;
192  end;

```

```

193 while again do
194   begin
195     titel;
196     escape:=false;
197     schriftart;
198     zeichensatz;
199     zeilenzahl;
200     abstand;
201     kursiv;
202     doppelt;
203     if escape then again:=false
204     else begin
205       writeln;writeln;
206       writeln(blk,'Angaben korrekt ? (j/n)',nblk);
207       read(kbd,zeich);
208       again:=zeich='n';
209     end;
210   end;
211   clrscr;
212   write(con);
213 end.
214

```

Bedienungshinweise zu SETLST

=====

1) SETLST fragt zunaechst den Drucker auf Bereitschaft zur Uebernahme der Daten ab. Ergibt sich dabei ein nicht betriebsbereiter Zustand, so wartet SETLST auf den Eintritt eines solchen. (indem man z.B. den Drucker einschaltet). Aus dieser Warterei kann SETLST jedoch auch durch Druck auf eine bel. Taste 'erloest' werden, was den Abbruch des Programms bewirkt.

2) Wird Druckerbereitschaft festgestellt, so fragt SETLST den Wert der einstellbaren Parameter vom Benutzer ab. Dieser hat folgende Reaktionsmoeglichkeiten:

'->'-Taste: waehle rechten Wert

'<-'-Taste: waehle linken Wert

RET -Taste: uebertrage den gewaehlten Wert an den Drucker
(der jeweils gewaehlte Wert wird invers dargestellt)

ESC -Taste: uebertrage den gewaehlten Wert an den Drucker und uebertrage fuer die noch folgenden Parameter automatisch die voreingestellten Werte

```
1  program mini_piano;
2
3  { Beispielprogramm fuer Definition und Anwendung elementarer
4    Routinen zur Tonerzeugung unter Turbo-Pascal
5    von Wolfgang Tesch    07.08.1985
6  }
7
8  type frequenz = 0..1023;
9       lautst   = 0..15;
10      kanal    = (I,II,III,R);
11      tonkanal = I..III;
12      shiftrate= 0..3;
13      r_art    = (p,w);
14
15  function hz (tonhoehe:integer) : frequenz;
16  { erlaubt die Eingabe eines Tonhoehenwertes in Hz
17    und wandelt ihn in den entsprechenden Wert zur
18    Frequenzangabe fuer den MTX-Tongenerator }
19
20  const takt = 4E+06;
21  begin
22    hz := trunc ( takt / 32/tonhoehe );
23  end;
24
25
26  procedure ton ( k:tonkanal ; f:frequenz ; l:lautst );
27  { erzeugt einen (Dauer-) Ton }
28
29  var dummy : byte;
30  begin
31    port[6]:= 128+(16*ord(k)*2)+(f mod 16);
32    dummy := port[3];
33    port[6]:= f div 16;
34    dummy := port[3];
35    port[6]:= 128+(16*(ord(k)*2+1))+15-l;
36    dummy := port[3];
37  end;
38
39
40  procedure rauschen ( sr:shiftrate ; a:r_art ; l:lautst );
41  { erzeugt Rauschen }
42
43  var dummy : byte;
44  begin
45    port [6] := 224 + ord(a)*4 + sr;
46    dummy:=port[3];
47    port [6] := 240 + 15-l;
48    dummy := port[3];
49  end;
50
51
52  procedure aus ( k:kanal);
53  { Lautstaerke des betreffenden Kanals auf 0 }
54
55  var dummy : byte;
56  begin
57    port[6]:=128 + 16*(2*ord(k)+1) + 15 ;
58    dummy := port[3];
59  end;
60
61
62
63  procedure stille;
64
```

```

65
66 var k:kanal;
67 begin
68   for k := I to R do
69     aus(k);
70 end;
71
72 { ----- }
73 { Die folgenden Programmteile erzeugen ein Beispiel zur Anwendung der
74   oben aufgelisteten Tonerzeugungs-Proceduren }
75
76 procedure titelbild;
77
78 procedure zeichne_tasten;
79   type leerstr = string[10];
80       instr = string[10];
81       outstr = string[14];
82   const leer: leerstr = '          ';
83   var i:byte;
84
85   function rev (text:instr) : outstr;
86     const revaus = ^F#4;
87           revan = ^F#16;
88   begin
89     rev:=revan+text+revaus;
90   end;
91
92   procedure line_1;
93   begin
94     write (leer);
95     for i:= 1 to 2 do
96       write ('! ',rev('  '),', ',rev('  '),', ' | ',rev('  '),
97             ', ',rev('  '),', ',rev('  '),', ' ');
98     writeln ('! ');
99   end;
100
101   procedure line_2;
102   begin
103     write(leer);
104     writeln('! ',rev(' s '),', ',rev(' d '),', ' | ',rev(' g '),', ',
105            rev(' h '),', ',rev(' j '),', ' | ',rev(' 2 '),', ',
106            rev(' 3 '),', ' | ',rev(' 5 '),', ',rev(' 6 '),', ',
107            rev(' 7 '),', ' | ');
108   end;
109
110   procedure line_3;
111   begin
112     write(leer);
113     for i:= 1 to 15 do
114       write('! ');
115     writeln('!');
116   end;
117
118   procedure line_4;
119   type tas = string[15];
120   const t : tas = 'yxcvbnmqwertzui';
121   begin
122     write(leer);
123     for i:= 1 to 15 do
124       write('! ',t[i],', ');
125     writeln('!');
126   end;
127
128   procedure line_5;

```

```

129     begin
130         write(leer);
131         for i:= 1 to 15 do
132             write('!___');
133             writeln('!');
134     end;
135
136     begin
137         gotoxy(21,3);
138         write('*****');
139         gotoxy(21,4);
140         write('*                *');
141         gotoxy(21,5);
142         write('*          M I N I - P I A N O          *');
143         gotoxy(21,6);
144         write('*                *');
145         gotoxy(21,7);
146         writeln('*****');
147         writeln;writeln;
148         line_1; line_1; line_1;
149         line_2;
150         line_3; line_3; line_3;
151         line_4;
152         line_5;
153         writeln;writeln;
154         writeln(leer,' zum Beenden des Programms --> ESC-Taste druecken');
155     end;
156
157     begin
158         zeichne_tasten;
159     end;
160
161
162
163     procedure klangerzeugung;
164     type keyb = array[0..24] of char;
165     const taste : keyb = ('y','s','x','d','c','v',
166                          'g','b','h','n','j','m',
167                          'q','z','w','3','e','r',
168                          '5','t','6','z','7','u','i');
169     type toene = array[1..25] of integer;
170     const note : toene = ( 261,277,294,311,330,349,
171                          370,392,415,440,466,494,
172                          523,554,587,622,659,698,740,
173                          784,831,880,932,988,1046);
174
175     var ende,gef : boolean;
176         ind : 0..25;
177         ch : char;
178
179     procedure fadeout;
180     var ls:lautst;
181         hilf:byte;
182     begin
183         ls:=13;
184         repeat
185             ls:=ls-1;
186             ton(I,HZ(note[ind]),ls);
187             ton(II,HZ(note[ind]),ls);
188             ton(III,HZ(note[ind]*2),ls);
189             hilf:=90;
190             repeat
191                 hilf:=hilf-1;
192

```

```
193     until keypressed or (ls=0)
194     end;
195
196
197 begin
198     ende:=false;
199     while not ende do
200         begin
201             read (kbd,ch);
202             if ch=#27 then ende:=true;
203             ind:=0;
204             gef:=false;
205             while (not gef) and (ind<25) do
206                 begin
207                     gef:=taste[ind] = ch;
208                     ind := ind+1;
209                 end;
210             if gef then
211                 begin
212                     ton(I ,Hz(note[ind]) ,12);
213                     ton(II ,Hz(note[ind] ),12);
214                     ton(III,Hz(note[ind]*2),12);
215                     fadeout;
216                 end;
217             end;
218         end;
219     begin
220         stille;
221         clrscr;
222         titelbild;
223         klangerzeugung;
224     end .
```

FRANK
BUESCHLER

Am Ochsenzoll 3
2000 Norderstedt
(040) 527 75 81

Frank Bueschler Am Ochsenzoll 3 2000 Norderstedt

An die Mitglieder des
MTX User-Clubs Deutschland

Computergasse 12

4711 Memotech

ca. Fenster eines Briefumschlags

Norderstedt, 20.09.1985

Liebe Mitglieder,

hier also eine Lösungsmöglichkeit für einen Briefkopf, der mit der "alten" VOBIS-NewWord Version erstellt wurde. Angeschlossen ist der DMX80, im Printermenue installiert als Epson MX80.

Den Briefkopf kann man abspeichern und dann in zu erstellende Briefe mit ^KR einladen.

Das entsprechende "Progrämmchen" findet Ihr auf der nächsten Seite.

Noch ein Hinweis zu dem in Info 6 Seite 40 beschriebenen NewWord-Patching von Peter Würfel. Bei dieser Installation muß der Drucker als DRAFT-Printer installiert sein. Also kann man dort beschriebene Schriftarten nicht in dem Brief nutzen.

Peter hatte das Problem, daß er im Schrifttyp "Elite" nur 80 Zeichen pro Zeile haben möchte --> PRINTING WIDTH DESIGNATION :
LPRINT CHR\$(27);"Q";(80);

Frank Bueschler

NewWord - Briefkopf für DMX80

(Frank Bueschler)

0.	.xq1B5000	Einstellung auf Elite
1.	.pl67	setzt Seitenlänge auf 67 Zeilen fest
2.	.pfon	page formatting on
3.	.mt0	Blattrand 0
4.	.lm2	linker Rand 2
5.	.ls0	line spacing 0
6.	^A ^Q	^PA auf altern. u. ^PB elite Schrift
7.	.cv34	setzt dieses auf Schriftbreite 34
8.	^N	^PN auf normalen Schriftsatz
9.	.cv5	setzt diesen auf Schriftbreite 5
10.	^A	^PA auf alternativen Schriftsatz
11.	.ls1	line spacing 1
12.	^BFRANK	^KB-Doppeldruck / Vorname
13.	BUESCHLER^N^B_.....^N	Name/^PN/^PB/_ bei mir 109 Mal/^PN
14.	.lm1	linker Rand 1
15.	.cv14	Schriftbreite 14
16.	.pfoff	page formatting off
17.	.mt3	oberer Blattrand 3
18.	^D^B	
19.		Am Ochsenzoll 3
20.		2000 Norderstedt
21.	.cv6	(040) 527 75 81^B^D^A Schriftbreite 6
22.		
23.	Frank Bueschler Am Ochsenzoll 3 2000 Norderstedt^N^D^B	
24.	.lm4	linker Rand 4
25.		
26.	An	
27.	*	
28.	*	
29.	*	
30.		
31.	*** *	
32.		
33.	.lm0	linker Rand 0
34.		
35.		
36.	^D	
37.	-	<u>Knickkante</u> für Fensterbriefe
38.		
39.		Norderstedt, **.**.***
40.	.lm5	linker Rand 5
41.	.rm70	rechter Rand 70
42.		
43.	Sehr geehrte Damen und Herren,	

Wo der linke Rand für die Textstellen sitzen muß, müßt Ihr ausprobieren, da Eure Adressen sicherlich andere Zeichenlängen haben. Sämtliche Schriftbreiten, die unter .cv7 liegen, kann der Drucker nicht in einem Zug drucken, daß heißt, er braucht für diese Zeichen etwas länger.

Ansonsten wünsche ich viel Spaß beim basteln, denn zur Erstellung dieser kleinen Routine habe ich mal mehrere Nächte gebraucht.

Im nächsten Info kommt eine ähnlicher Briefkopf für das neue NewWord von Herbert Herberg

Frank + Herbert

naturlich kein "Gewehr"..... Viel Spass : Frank Bueschler

Radio Schossau	Essen	0201/237396	22.00 - 10.00	
Mailbox Service	Essen	0201/274625	20.00 - 13.00	
R.M.S.	Essen	0201/422676	18.00 - ???	Sysop.CHRIS
Hanske's Mailbox	Essen	0201/736566		
Datex-P	Essen	0201/787051		
Ronsdorfer Box		0202/463678		Sysop.HORST
Toelleturm		0202/559350		Sysop.JUERGEN + KERSTIN
Mercator	Duisburg	0203/782497	22.00 - 08.00	Sysop.RAINER
R.A.F.-Box	Duisburg	0208/401763	20.00 - 06.00	
HPA-Box	Oberhausen	0208/845974	18.00 - 06.00	Sysop.HOLGER
Multi-Box		0209/271666		Sysop.KLAUS
R.T.C.	Ratingen	02102/68187	16.30 - 06.00	Sysop.HELMUT
E.V.D./RCPM	Düsseldorf	0211/328249		
Datex-P	Düsseldorf	0211/329318		
Data-Becker Mailbox	Düsseldorf	0211/340071		
Die 'Kiste'	Düsseldorf	0211/350180		nur kommerzieller An-/Verkauf
Swiss Mailbox	Düsseldorf	0211/407409		Sysop.SIR HACK
Software-Express	Düsseldorf	0211/414579		
Sharp	Düsseldorf	0211/452052		
Epson	Düsseldorf	0211/593453		
Nanco	Düsseldorf	0211/622350	18.00 - 06.00	
H.S.C.		02129/52883		
Lion-Soft	Krefeld	02151/700253	22.00 - 06.00	Mailboxsoftware
SYMIC	Mönchengladbach	02161/200928		kommerzieller Anbieter
Computer-Center	Berg.-Gladbach	02202/50033		kommerzieller Anbieter
Kenso's Creative		02204/57025		Sysop.KENSO
MEGBDX		02204/61806	16.00 - 19.00	
Saturn	Köln	0221/1616284		kommerzielles System
Underground Data	Köln	0221/236534		unter UDD5/1.3 - no public
Datex-P	Köln	0221/2911		
WDR (KOM-KOM) III.Pr	Köln	0221/371076		Sysop.GUENTHER
PMS (PersMessageSer)	Köln	0221/394976		Sysop.GERD
Mailbox	Köln	0221/4971929		
Koppler Treff	Köln	0221/503759	22.00 - 06.00	
Hacker Box	Köln	0221/512640		Sysop.HANS
K.C.T.-Bit Dschungel	Köln	0221/558336		Sysop.PETER
F.I.S.		02234/58603	20.00 - 10.00	Sysop.KLAUS
Datex-P		02234/78999		
HPF	Bonn	0228/211737	12.00 - 18.00	
S.I.S.	Schwerte	02304/44770		Sysop.UWE
Mailbox	Dortmund	0231/170414		
IBM	Dortmund	0231/441103		
Satellit 64	Dortmund	0231/441103		
Tremonia	Dortmund	0231/524968	20.00 - 07.00	
Datex-P	Dortmund	0231/57011		
CBBS	Dortmund	0231/650786		
IBM 370	Dortmund	0231/7552541		US:FINIX15 PA:THEOBALD EN:LOGON
Mythos	Dortmund	0231/779620		
G.L.A.T.Z.E.	Dortmund	0231/893445	20.00 - 06.00	Sysop.GOETTNER

Modem-Liste

Page 02

Kobra	Hagen	02331/16401		Sysop.UWE
C-64 Usergroup		02361/72928	22.00 - 10.00	
HID	Haltern	02364/13826	1.00 - 06.00	Sysop.UDD
UEDING-electronic		02373/66877	17.00 - 09.00	kommerzieller Anbieter
I.G.S. Sharp	Kamen	02383/50866		
TH	Aachen	0241/81081		
A.I.S.	Aachen	0241/870555	20.00 - 09.00	Sysop.AXEL
ALLDAS		0243/23602		Sysop.BERND
Mailbox		0261/200928		Sysop.GEORG
Moerser Hacker Box		02841/57325	22.00 - 06.00	Sysop.GEORG
Esprit		02841/66241		
Kunisoft Mailbox		02932/37748	22.00 - 12.00	Sysop.Kunisoft
Datex-P	Berlin	030/240001		
Berliner Mailbox	Berlin	030/3052635	18.00 - 09.00	
UNI	Berlin	030/314730		
Chaos Box	Berlin	030/3448756	22.00 - ???	
KKB	Berlin	030/3958673	19.00 - 23.00	
Garelli	Berlin	030/4144935	16.00 - 22.00	
Coko	Berlin	030/4652439	22.00 - 09.00	
NBCMB	Berlin	030/4926643	19.00 - 09.00	
Telemail	Berlin	030/4926643	19.00 - ???	
CB-Funk-Box	Berlin	030/66399596	19.00 - ???	
IBB	Berlin	030/6818679	22.00 - 09.00	
T.I.C.	Berlin	030/7115078		
Testline	Berlin	030/7214446	20.00 - 24.00	
Bert	Berlin	030/7466850		
M.C.C.	Berlin	030/7823350	15.00 - 22.00	
C.G.S.	Berlin	030/7868178	16.00 - 22.00	
Katy's COMPAC 64	Berlin	030/8024228		
Daily Mail	Berlin	030/8922239	20.00 - 07.00	
C.C.C.	Hamburg	040/364347702	22.00 - 06.00	
Max-Planck-Institut	Hamburg	040/4114269		-richtig ausloggen-besetzt?-dann auflegen-
UNI	Hamburg	040/41233098	20.00 - 06.00	
Datex-P	Hamburg	040/441231		
H.I.S.	Hamburg	040/4916117		
W-W-S	Hamburg	040/5246387	20.00 - 06.00	
Tornado	Hamburg	040/5277016		
C.L.I.N.S.H.	Hamburg	040/6306262		
M.C.S. (CCC-Info)	Hamburg	040/6523486		
H.M.S.	Hamburg	040/73113293		
C-64 User Club	Hamburg	040/7540598	20.00 - 23.00	
CCS	Hamburg	040/7868178	16.00 - 22.00	
RAM	Hamburg	040/8802383		
Wang-Info	Pinneberg	04101/23789		
SMC	Stade	04141/2386	19.00 - 22.00	
Datex-P	Bremen	0421/310131		
UNI	Kiel	0431/8804556		LOGIN erwünscht
N.C.S.	Kiel	04348/7513		
RRZM	Hannover	0511/2100611		
Datex-P	Hannover	0511/326651		
AG MC	Hannover	0511/401025		
Dr.Jones	Hannover	0511/884421		
Aquila	Hildesheim	05121/35146		

Modem-Liste

Page 03

Datex-P	Bielefeld	0521/59011		
C A T	Wolfsburg	05361/23353		
Taunus Mailbox		06081/9677		
Decates		06154/51433		
Otis		06181/48884		
Thor		06187/25828		
Datex-P	Mannheim	0621/409085		
Sambo	Mannheim	0621/682722		
C.C.C.C.	Camberg	06434/6291		
Datex-P	Saarbrücken	0681/810011		
Hobby-C 1		06826/2234	15.00 - 19.00	
Hobby-C 2		06826/6344	19.00 - 23.00	
Datex-P	Frankfurt	069/20281		
A.U.G.E. (Apple)	Frankfurt	069/494201	19.00 - 06.00	
Control-Data	Frankfurt	069/63501110-29		
COMBO (Commodore)	Frankfurt	069/6638191		
M.R.N	Frankfurt	069/724513		
FIB	Frankfurt	069/726527	18.00 - 22.00	
Tecos	Frankfurt	069/816787	17.00 - 07.00	
IBM	Frankfurt	069/835039	19.00 - 21.30	
Box Vaihingen		07024/24205		
ELIAS		07031/278296	17.30 - 22.00	
Datex-P	Stuttgart	0711/299171		
PFM	Stuttgart	0711/461032	19.00 - 06.00	
NORSAK	Stuttgart	0711/519008		LOGIN:Pluto
Coloss		07191/62753	21.00 - 24.00	
Sputnik		07191/85784	21.00 - 06.00	
Datex-P	Karlsruhe	0721/60241		
M.C.S.	Karlsruhe	0721/685010		
Eisinger Infoservice		07232/81278		
Freiburger Box		0761/86705	20.00 - 06.00	
Microsoft		0791/42919		
Datex-P	Augsburg	0821/464011		
Info-Control KG	München	089/164959		
Mailbox	München	089/222066		
Datex-P	München	089/228730		
Graphon	München	089/320952		
LOGIN 11,3	München	089/32095264		help
CF-Computer	München	089/557318		
TEDAS I	München	089/596422		Franzis-Verlag
TEDAS II	München	089/598423		Franzis-Verlag
Mailbox	München	089/831564		
Orbit	München	089/9036130	23.00 - 06.00	
Datex-P	Nürnberg	0911/20571		
UBN	Nürnberg	0911/334927	17.00 - 24.00	
Smurf-O-Box	Darmstadt	0911/574160	18.00 - ???	
Mailhouse		09363/5329	19.00 - 09.00	

Was wir fast vergaßen

Am Freitag waren die Infos in der Druckerei, und es fehlten die folgenden Seiten, und ein paar der Dinge die hier weiter unten stehen. Also habe ich Samstag in der Druckerei angerufen und gefragt, ob ich Montag noch ein paar Seiten nachreichen könnte - offensichtlich mit positiver Antwort! Mein Gott wäre das peinlich! Da würden doch glatt die vielen Assemblerbefehle fehlen.

Public-Domain - CP/M-Gratissoftware

An was für CP/M-Software seid Ihr interessiert. Bitte schreibt mir das möglichst rasch, damit wir die Disketten bestellen und weitergeben können. Zur Zeit haben wir sechs Disketten mit dBase-Programmen. Im nächsten Info seht dann was von wem zu haben ist.

Mögliche Themen: MBASIC, Spiele, Assembler, Dienstprogramme,

Thema FORMULARE

Wenn wir uns schon die Mühe machen, und sogar mal Formulare für irgendetwas entwerfen, dann erwarten wir auch, daß Ihr die verwendet!!! Wer also solche vorhandenen Formulare (Anmeldung zu Clubtreffen, Bestellzettel, ...) nicht verwendet, wird ignoriert. Für uns ist das dann nämlich ein unnötiger Zusatzaufwand, weil die gesuchten Informationen an anderer Stelle stehen, und damit das schnelle blättern weniger bringt. Also **bitte** macht uns das Leben leichter, indem Ihr solche Formulare verwendet!

PAL / PROM / EPROM / RAM / ...

Was ist was? Nun möchte ich da versuchen etwas Aufklärung zu betreiben. Wenn also ein PAL mit einem PROM - naja, Ihr wißt schon - dann gibt's

PAL = Programmable Array Logic = Programmierbares Logik-Gatter, das ist ein IC mit verschiedenen Logik-Gattern, von denen man einige außer Betrieb setzen kann um so die gewünschte Funktion des IC's zu erhalten.

PROM = Programmable Read Only Memory = Programmierbarer Lesespeicher, ist ein Speicher, der einmal mit einem Inhalt beschrieben werden kann, und diesen für immer behält.

PAL und PROM sind sich also relativ ähnlich in dem Sinne, daß beim Programmieren (auch Brennen genannt) verschiedenen Kombinationen von Eingangssignalen bestimmte Ausgangssignale zugeordnet werden können. Der Hauptunterschied ist der, daß PAL's nicht so vielfältig programmiert werden können wie PROM's.

EPROM = Erasable PROM = Löschbares (also unprogrammierbares) PROM

RAM = Random Access Memory = Schreib-Lese-Speicher

D. Weineck:

Ich habe noch ein MTX-500 Grundgerät sowie einen Seikosha Drucker bei mir herumstehen. Würdest du für mich eine Anzeige in der Info drucken.

Ich bin Abend in der Zeit von 19-20 h zu erreichen

T e l 05208/7162

Preisvorstellung VHB 750.00 DM

Sorry, aber diese Seite musste ich entfernen, weil ich für eine Veröffentlichung außer in der ehemaligen, gedruckten Form keine Freigabe habe.